



GEORG-SIMON-OHM
HOCHSCHULE NÜRNBERG

Fakultät Elektrotechnik Feinwerktechnik Informationstechnik

Studiengang Software Engineering und Informationstechnik

Masterarbeit von Jens Thielemann

Konzeptionierung eines drahtlosen Elektrokardiogramm-Monitors

WS 2009/2010

Betreuer: Prof. Dr. Holger Carl

Kurzbeschreibung

Die Überwachung der Herzfunktion mittels Elektrokardiogramm (EKG) ist eine wichtige und unter Umständen lebensrettende Massnahme. Im stationären bzw. ambulanten Umfeld gilt sie schon lang als ein Standard-Diagnose-Verfahren.

Patienten, bei denen die Herzbeschwerden nicht permanent auftreten sondern eher sporadischer Natur sind, müssen über einen längeren Zeitraum überwacht werden. Herzbeschwerden dieser Art treten oft als Herzrhythmusstörungen in Erscheinung.

Im Rahmen dieser Masterarbeit wird ein drahtloser EKG-Monitor konzipiert, der die EKG-Signale des Patienten aufzeichnet und diese gegebenenfalls einem Arzt zur Diagnose übermitteln kann.

Im Fokus der Arbeit steht die Signalaufbereitung und die Datenübertragung des EKG-Signals. Die Arbeit beginnt mit der Vorstellung des Systemkontextes der Monitoranwendung. Nach der Beschreibung des internen Aufbaus des Monitors, folgt eine Darstellung der Herzfunktion und der damit verbundenen Entstehung des EKG-Signals samt Erläuterung der EKG-Abschnitte. Zur Aufbereitung des Signals werden verschiedene Filterkonzepte vorgestellt und in Matlab simuliert. Um das nun aufbereitete Signal effektiv übertragen zu können, folgt eine Abhandlung von Techniken zur Datenreduktion und Codierung. Die Simulation von Übertragungsfehlern zusammen mit der Codierung erfolgt mittels eines dafür entwickelten Qt-Programms unter Windows. Die Beschreibung zur Detektion des EKG-Signals rundet die Arbeit ab. Die Algorithmen zur Signalaufbereitung, Teile der Datenreduktion und der Signaldetektion werden in einem Simulationsmodell unter Matlab realisiert.

Abstract

The survey of the heart function by Electrocardiogram (ECG) is an important and potentially live-saving measure. In the stationary and ambulatory Environment it is used long ago as a standard diagnostic method.

Patients who suffer under heart trouble not permanently but rather sporadically, have to be surveyed over a longer time period. Heart trouble of that kind are often known as disturbance of the heart rhythm or arrhythmia.

In line with this masterthesis, a wireless ECG monitor will be designed, which records ECG signals of a patient and probably sends it to a physician for diagnostic issues.

The focus of the thesis is the signalprocessing and the datatransmission of ECG signals.

The thesis starts with the introduction of the ECG monitor system context. After describing the internal structure of the monitor a depiction of heart function follows with the ECG signal origin description and exposition of the ECG sections. For signalprocessing different filter concepts will be introduced and simulated in matlab. To transmitt the signal effectively, a disquistion about methods for datareduction and coding follows. The simulation of transmission errors together with the coding happens using by a Qt program developed under windows. The describtion for ECG signal detection rounds off the thesis. The algorithms for signalprocessing, parts of datareduction and signaldetection will be implemented in a simulation model under matlab.

Eidesstattliche Erklärung

Ich bestätige, dass ich die Abschlussarbeit mit dem Titel

Konzeptionierung eines drahtlosen EKG-Monitors

selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt, sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Nürnberg, den 26. September 2009

JENS THIELEMANN

Inhaltsverzeichnis

Formelzeichen und Abkürzungen	IV
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	XI
Verzeichnis der Listings	XII
1. Einleitung	1
1.1. Ziel der Arbeit	1
1.2. Aufbau der Arbeit	1
1.3. Voraussetzungen zum Verständnis der Arbeit	2
1.4. Motivation	3
2. Systemkontext	4
2.1. Verteilung	4
2.2. Nahfeldkommunikation	5
2.2.1. Bluetooth	5
2.2.2. Simple Serial Wireless Connection	7
2.3. Fernfeldkommunikation	9
2.4. Resümee	10
3. Aufbau	12
3.1. Übersicht	12
3.2. Signalquelle	14
3.3. Signalaufbereitung	16
3.4. Datenkompression	17
3.5. Kanalcodierung	19
3.6. Detektion	21
3.7. Domain-Modell	21
4. Elektrokardiogramm	23
4.1. Entstehung	23

4.2. EKG-Abschnitte	24
4.3. Frequenzspektrum	26
4.4. EKG-Generator	28
4.5. Resümee	30
5. Filterkonzepte	31
5.1. Anforderungen	31
5.2. Mittelwertschwankungen	33
5.3. Schätzung von Signalwerten	34
5.3.1. Autokorrelation	34
5.3.2. Wiener-Hopf-Gleichung	36
5.3.3. Levinson-Durbin-Algorithmus	37
5.3.4. Autoregressive Spektralschätzung	38
5.4. Rauschunterdrückung	40
5.5. Dämpfung höherfrequenter Signalanteile	42
5.5.1. Filterspezifikation	42
5.5.2. Filterentwurf	43
5.6. Blockschaltbild	45
6. EKG-Signalkompression	47
6.1. Dekorrelation	47
6.1.1. Lineare Prädiktion	47
6.1.2. Prädiktion mit Quantisierung	49
6.2. Entropiecodierung	52
6.2.1. Huffman-Codierung	52
6.2.2. Codegenerierung	53
6.3. Blockschaltbild	56
7. Datenübertragung	58
7.1. Bitfehlerwahrscheinlichkeit	58
7.2. Kanalmatrix	59
7.3. Blockcode-Generierung	61
7.4. Datenprotokoll	64
7.5. Blockcode-Decodierung	65
7.6. Simulator	66
7.7. Resümee	68
8. Signalmrückgewinnung	69
8.1. Blockschaltbild	69

8.2. Huffman-Decoder	70
8.3. Prädiktor-Decodierung	73
8.4. Interpolation	74
8.5. Resümee	76
9. Signaldetektion	77
9.1. Algorithmus	77
9.2. Beispiele	80
9.2.1. Herzrhythmusstörungen	80
9.2.2. Vorderwandinfarkt	82
9.2.3. Fehlerhafte Übertragung	83
9.3. Resümee	86
10. Zusammenfassung und Ausblick	88
Literaturverzeichnis	90
A. Anhang	i
A.1. Listing zum EKG-Signalspektrum	ii
A.2. Listing zum Mittelwertfilter	iv
A.3. Listing zum Levinson-Durbin Algorithmus	v
A.4. Listing zur Rauschunterdrückung	vii
A.5. Listing zur Skalierung und Koeffizientenquantisierung des Tiefpassfilters . .	ix
A.6. Listing zur QRS-Komplex Detektion	xi
A.7. Simulator für Code Generierung und Übertragung	xii
A.7.1. Anleitung zur Bedienung des Simulators	xiii
A.8. Komponentendiagramme zur Software Umsetzung	xiv

Formelzeichen und Abkürzungen

a_i	IIR-Filterkoeffizienten
a_n	Reelle Fourierkoeffizienten
α_i	Vereinfachung für FIR-Filterkoeffizient $-b_{i-1}$
b	Bandbreite
b_i	FIR-Filterkoeffizienten
b_n	Imaginäre Fourierkoeffizienten
b_{opt}	Optimale Koeffizienten des Prädiktionsfilters
\underline{B}	Strukturelement Vektor der Maske B[m]
$\underline{c}_n(nf_0)$	Komplexe Fourierkoeffizienten
C	Kanalkapazität
d_{min}	Hammingdistanz
$\delta_{(s0,w)}^*$	Zustandsübergänge des Automaten
e_{det}	Anzahl erkennbare Einzelfehler
e_{kor}	Anzahl korrigierbare Einzelfehler
$e[n]$	Prädiktionsfehler
$e_Q[n]$	Quantisierter Prädiktionsfehler
E_b	Energie pro Bit
$E(x)$	Erwartungswert von x
η	Codiereffizienz
f_s	Abtastfrequenz (Sampling frequency)
f_{stop}	Maximale Frequenz eines Signals
F	Endzustandsmenge des Automaten
$g(x)$	Generatorpolynom
G	Generatormatrix
G_{ks}	klar-systematische Generatormatrix
G_p	Prädiktionsgewinn
H	Prüfmatrix
$H(x)$	Entropie der Zeichen x
$H(z)$	Übertragungsfunktion im z-Bereich
$H_{AR}(e^{j\Omega})$	Übertragungsfunktion der autoregressiven Spektralschätzung im Frequenzbereich

$H_{AR}(z)$	Übertragungsfunktion der autoregressiven Spektralschätzung im z-Bereich
$H_e(z)$	Übertragungsfunktion des Prädiktionsfehlers im z-Bereich
$H_{max}(y)$	Maximale Entropie der Zeichen y
$H(y x)$	Streuentropie, Irrelevanz
$I(x_i)$	Informationsgehalt des Zeichens x_i
k	Boltzmann-Konstante
k_p	Reflexionskoeffizient für Lattice-Filter
L	Mittlere Codewortlänge
L_A	Sprache des Automaten
m_i	Codewortlänge des Zeichens x_i
mod	Modulo-Operation
MSE	Mittlerer quadratischer Fehler
μ_x	Erwartungswert
N_0	Spektrale Rauschleistungsdichte
$N(z)$	Weisses Rauschsignal im z-Bereich
p_b	Bitfehlerwahrscheinlichkeit
p_{Rdet}	Restfehlerwahrscheinlichkeit
p_{RdetBF}	Restfehlerwahrscheinlichkeit für Bündelfehler
$p(x_i)$	Auftretenswahrscheinlichkeit des Zeichens x_i
$p(y x)$	Kanalmatrix
P_N	Rauschleistung
P_S	Signalleistung
r	Empfangscodewort
r_i	Kurzschreibweise für Autokorrelationsfunktion $R_{xx}[i]$
r_{xx}	Vektor der Autokorrelationsfunktion
R	Coderate
$R_{ee}[l]$	Autokorrelationsfunktion von $e[n]$
R_{xx}	Symmetrische Toeplitz-Matrix der Autokorrelationsfunktion
$R_{xx}[l]$	Autokorrelationsfunktion von $x[n]$
ρ_x^2	quadratischer Mittelwert (zeitkontinuierliche Signale)
$\rho_x^2[n]$	quadratischer Mittelwert (zeitdiskrete Signale)
s	Syndrom
$s(t)$	Zeitkontinuierliches Signal
S	Zustandsmenge des Automaten
$\underline{S}(f)$	Spektralfunktion
$\hat{S}_{AR}(e^{j\Omega})$	Ausgangsleistungsdichtespektrum der autoregressiven Spektralschätzung

Formelzeichen und Abkürzungen

SNR	Signalrauschverhältnis
σ_e^2	Varianz des Prädiktionsfehlersignals
σ_N^2	Varianz des Weissen Rauschsignals
σ_x^2	Varianz zeitkontinuierlicher Signale
$\sigma_x^2[n]$	Varianz zeitdiskreter Signale
T_0	Raumtemperatur
T_Z	Empfängertemperatur
w	Wort des Automaten
x_i	Zeichen x_i
$x_Q[n]$	Quantisierte Werte von $x[n]$
$\hat{x}[n]$	Schätzwerte von x
\underline{X}	Datenvektor der Zeichen $X[n]$
X	Zeichenmenge des Automaten
$y[n]$	Zeitdiskretes Ausgangssignal
$Y_{AR}(z)$	Ausgangssignal der autoregressiven Spektralschätzung im z -Bereich
AKF	Autokorrelationsfunktion
AR	Autoregressive model
ARQ	Automatic Repeat Request
ASK	Amplitude Shift Keying
BPMC	BiPhase Mark Coding
BSC	Binary Symmetric Channel
CRC	Cyclic Redundancy Check
EKG	Elektrokardiogramm
EMV	Elektromagnetische Verträglichkeit
FEC	Forward Error Correction
FIR	Finite Impulse Response
FM	Frequenzmodulation
FSK	Frequency Shift Keying
GPS	Global Positioning System
GSM	Global System for Mobile communication
HFP	Handsfree Profile
HRV	Herzratenvariabilität
HW	Hardware
IIR	Infinite Impulse Response
ISM	Industrial, Scientific and Medical Hochfrequenzband
L2CAP	Logical Link Control And Adaption

MOST	Media Oriented Systems Transport
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error (Mittlerer Quadratischer Fehler)
MVC	Model View Controller pattern
NLOS	None Light Of Sight (Keine Sichtverbindung zwischen Sender und Empfänger)
FHSS	Frequency Hopping Spread Spectrum (Wechsel der Trägerfrequenz durch Pseudozufallszahlen)
RFCOMM	Radio Frequency Communication
SIG	Special Interest Group (Konsortium zur Ausarbeitung eines Bluetooth Standards)
SIMAP	SIM Access Profile
SMS	Short Message Service
SNR	Signal to Noise Ration
SPP	Serial Port Profile
SSWC	Simple Serial Wireless Connection (einfache Realisierung einer seriellen Funkstrecke)
UML	Unified Modelling Language

Abbildungsverzeichnis

1.1. Statistik von Herzerkrankungen in Deutschland	3
2.1. Verteilung der Komponenten	5
2.2. Bluetooth Protokoll Stack (Bildquelle: [Merkle 2002] modifiziert)	7
2.3. Simple Serial Wireless Connection Schichtenmodell	8
2.4. Empfänger zur Datenübertragung mit SSWC, 1) FM-Modul, 2) Prozessor für BPMC	9
2.5. Notruffunktionalität	10
3.1. Modell der Verarbeitungskette (Bildquelle: [Strutz 2005] modifiziert)	12
3.2. Übersicht der in der Monitor Unit eingesetzten Funktionsblöcke	13
3.3. Einthoven-Ableitungen (Bildquelle: [IfBTUK 2009])	14
3.4. Nehb Ableitung (Bildquelle: [IfBTUK 2009])	15
3.5. EKG Folie (Bildquelle: [FIPM 2009])	16
3.6. Domain-Modell für die verwendeten Techniken	21
4.1. Entstehung des EKG-Signals (Bildquelle: [EKGEntstehung 2009] modifi- ziert), 1) Sinusknoten, 2) AV-Knoten, 3) Mitralklappe, 4) Trikuspidalklap- pe, 5) Aortenklappe und dahinter Pulmonalklappe	24
4.2. Abschnitte des EKG-Signals, 1) Erregung der Vorhöfe, 2) Kammererregung, 3) Erregungsrückbildung	25
4.3. EKG-Pulsabschnitt aus Zeitfunktionen zusammengesetzt	26
4.4. Linienspektren der modellierten EKG-Pulsabschnitte	28
4.5. EKG-Pulsabschnitt simuliert	29
4.6. Spektrum EKG-Pulsabschnitt simuliert mit einer Herzfrequenz von 120 Schlägen pro Minute	29
5.1. Blockschaltbild zur Veranschaulichung der Anforderungen an das Filterkon- zept	31
5.2. Frequenzspektrum des EKG-Signals überlagert mit Störsignalen	32
5.3. Mittelwertschwankungen des EKG-Signals	33
5.4. Filter zur Kompensation von Schwankungen der Nulllinie (Bildquelle: [Kris- hnan 2007] modifiziert)	33

5.5. Prinzipieller Aufbau des Medianfilters (Bildquelle: [Krishnan 2007] modifiziert)	34
5.6. Kompensation der Schwankungen der Nulllinie des EKG-Signals	34
5.7. Autokorrelationsfunktion des EKG-Signals	35
5.8. Generierung von Filterkoeffizienten zur Schätzung von Signalwerten	38
5.9. Blockschaltbild des Filters zur AR Modellierung	39
5.10. Arbeitsweise der Filterung mit Morphologischen Operatoren	41
5.11. Rauschunterdrückung des gestörten EKG-Signals mittels morphologischen Operatoren. Oben das Signal vor der Filterung. Unten das Signal nach der Filterung	42
5.12. Matlabs FDATool zur Filterspezifikation	43
5.13. Filter 8. Ordnung mit Blöcken zweiter Ordnung in Direktform-II-Struktur (Bildquelle: [Grueningen 2002] modifiziert)	43
5.14. Übertragungsfunktion des Tiefpassfilters	44
5.15. Pol-, Nullstellendiagramm des Tiefpassfilters	44
5.16. Blockschaltbild der Signalaufbereitung	45
5.17. Frequenzspektrum nach der Signalaufbereitung	46
6.1. Blockschaltbild für die lineare Prädiktion	48
6.2. Blockschaltbild für die lineare Prädiktion mit Quantisierung	49
6.3. Prädiktionsfehler mit Prädiktion 2.Ordnung	50
6.4. Prädiktionsfehler mit Prädiktion 5.Ordnung	50
6.5. Abhängigkeit des MMSE-optimalen Prädiktors 2.Ordnung von seinen Koeffizienten	52
6.6. Häufigkeitsverteilung der zu codierenden Zeichen	53
6.7. Schritte zur Erzeugung eines Huffman-Codebaumes	54
6.8. Anzeige der Code-Tabelle des Simulators	55
6.9. Blockschaltbild der Datenkompression	56
7.1. Bitfehlerwahrscheinlichkeiten bipolarer und unipolarer Übertragung	59
7.2. Kanalmatrix des binären symmetrischen Kanals	59
7.3. Klassendiagramm des Simulators	67
8.1. Blockschaltbild zur Signalarückgewinnung	69
8.2. Binärbaum zur Decodierung der Huffman-Codes	70
8.3. Endlicher deterministischer Automat zur Decodierung der Huffman-Codes	71
8.4. Anzeige der decodierten Daten des Simulators	72
8.5. Prädiktion Decodierung mittels All-pole-Lattice-Filter (Bildquelle: [Werner 2008] modifiziert)	74

8.6. Frequenzspektrum vor der Datenkompression (Amplitude 1) und nach der Signalarückgewinnung (Amplitude 2)	76
9.1. Blockschaltbild zur Detektion der P-,Q-,R-,S-,T-Pulse	77
9.2. EKG-Signal (oben) und das Ausgangssignal des QRS-Detektors (unten) . .	78
9.3. Diagramm zur Detektion der P-,Q-,R-,S-,T-Pulse mit einer Herzfrequenz von 60 Schlägen pro Minute	79
9.4. Diagramm zur Detektion der P-,Q-,R-,S-,T-Pulse mit einer Herzfrequenz von 120 Schlägen pro Minute	79
9.5. EKG-Signal der Herzrhythmusstörung AV-Block 2. Grades Typ 2 (oben) und die Auswertung (unten)	81
9.6. EKG-Signal der Herzrhythmusstörung AV-Block 3. Grades (oben) und die Auswertung (unten)	82
9.7. EKG-Signal des Vorderwandinfarkts (oben) und die Auswertung (unten) . .	83
9.8. Simulatorengabe zur Erkennung von Bitfehlern (links) und zur Schätzung des Korrekturwertes (rechts)	84
9.9. Auswirkung des Bitfehlers auf die Auswertung	84
9.10. Simulation einer fehlerhaften Datenübertragung mit mehreren Bitfehlern . .	85
9.11. Auswirkung der Bitfehler auf die Auswertung	86
A.1. Haupt- und Ausgabefenster des Simulators	xii
A.2. Biterror-Dialogfenster	xiii
A.3. Komponentendiagramm für die Software Umsetzung des EKG-Monitor Sender	xiv
A.4. Komponentendiagramm für die Software Umsetzung des EKG-Monitor Empfangsteil	xv

Tabellenverzeichnis

2.1. Bluetooth-Reichweiten in Abhängigkeit der Klassen [Merkle 2002]	6
2.2. Gegenüberstellung von Bluetooth und SSWC	11
3.1. Einteilung von Techniken zur Verringerung der Datenmenge	17
3.2. Unterteilung der Verfahren zur Datenreduktion	18
3.3. Verfahren zur Dekorrelation von Signaldaten	18
3.4. Unterteilung der Verfahren zur Codierung	19
3.5. Codeklassen der Kanalcodierung	20
4.1. Normalwerte für Amplitude und Dauer der EKG-Pulse	25
5.1. Quantisierte Koeffizienten für das Filter mit Blöcken 2. Ordnung in Direktform-II-Struktur	45
6.1. Prädiktionsgewinn in Abhängigkeit der Prädiktorordnung	51
7.1. Codeworte in Polynom-Darstellung und in hexadezimaler Schreibweise . . .	63
8.1. FIR-Filterkoeffizienten zur Interpolation	75
9.1. Tabelle zur Darstellung der Übertragungsfehler und Schätzung der Korrekturwerte	85

Verzeichnis der Listings

A.1. Berechnung EKG-Signalspektrum in Matlab	ii
A.2. Mittelwertfilter in Matlab	iv
A.3. Levinson-Durbin Algorithmus zur Berechnung der Reflexionskoeffizienten in C	v
A.4. Rauschunterdrückung mittels morphologischen Operatoren in Matlab	vii
A.5. Skalierung und Koeffizientenquantifizierung des Tiefpassfilters in Matlab . .	ix
A.6. QRS-Komplex Detektion in Matlab	xi

1. Einleitung

1.1. Ziel der Arbeit

Im Mittelpunkt steht die Erstellung eines Simulationmodells mit Matlab als Konzeptionsgrundlage für den EKG-Monitor. Die eingesetzten Techniken zur EKG-Signalverarbeitung und Auswertung werden hierbei nach ihrer Funktion und Einsetzbarkeit hin untersucht und verifiziert. Darüber hinaus ist die Erstellung einer Simulation der Datenübertragung ebenso ein wichtiger Bestandteil der Arbeit. Hierbei gilt es die für die Datenübertragung eingesetzte Kanalcodierung zu realisieren. An dieser Stelle wird die Notwendigkeit der Realisierung mittels der Programmiersprache C/C++ hervorgehoben, um eine eventuell nachfolgende Implementierung auf eine HW-Plattform zu erleichtern.

1.2. Aufbau der Arbeit

Kapitel 1, Einleitung

Neben den Voraussetzungen zum Verständnis der Arbeit wird auf die Motivation der Arbeit eingegangen.

Kapitel 2, Systemkontext

Die Anwendung des Monitors wird hier erläutert. Ebenso werden die dafür in Frage kommenden Funkkonzepte diskutiert.

Kapitel 3, Aufbau

In diesem Kapitel erfolgt die Beschreibung des internen Aufbaus des Monitors. Begonnen bei der Signalquelle bis zur Detektion werden die Komponenten beschrieben. Ein Domain-Modell zur Übersicht der eingesetzten Techniken schließt die Beschreibung ab.

Kapitel 4, Elektrokardiogramm

Die Entstehung und Beschreibung der EKG-Abschnitte kann in diesem Kapitel nachgelesen werden. Ebenso erfolgt hier eine erste theoretische Betrachtung des Signalspektrums.

1. Einleitung

Kapitel 5, Filterkonzepte

Für die richtige Aufbereitung des Signals werden die Anforderungen dafür zusammengetragen. Anschließend folgen unterschiedliche Filterkonzeptionen und deren Grundlagen zur Störunterdrückung.

Kapitel 6, EKG-Signalkompression

Dieses Kapitel beschäftigt sich mit dem Thema der Reduzierung von EKG-Daten, um eine effiziente Datenübertragung zu ermöglichen.

Kapitel 7, Datenübertragung

Der Inhalt dieses Kapitels beschreibt die Art der Kanalcodierung und stellt ein Programm zur Simulation der Übertragung vor.

Kapitel 8, Signalmrückgewinnung

Damit das Signal später weiter verarbeitet werden kann, erfolgt in diesem Kapitel die Beschreibung, wie das Signal empfangen und decodiert wird.

Kapitel 9, Signaldetektion

Ein Algorithmus zur Detektion des EKG-Signals wird vorgestellt. Es folgen Beispiele, die die Wirksamkeit der Detektion demonstrieren.

Kapitel 10, Zusammenfassung und Ausblick

In diesem Kapitel erfolgt eine Zusammenfassung der Arbeit. Im Ausblick werden weiterführende Punkte basierend auf dem Ergebnis der Arbeit angesprochen.

Anhang

Dem Anhang können diverse Listings, eine Simulatorbeschreibung und Komponentendiagramme zu den besprochenen Themen entnommen werden.

1.3. Voraussetzungen zum Verständnis der Arbeit

Vorausgesetzt werden Grundlagen der digitalen Signalverarbeitung, der Informationstheorie und der Codierung. Die Ableitungen der EKG Signale vom menschlichen Körper, ebenso die Deutung von Herzkrankheiten können in dem Rahmen dieser Arbeit nur oberflächlich behandelt werden. Zur Vertiefung empfiehlt sich [So 2001].

Sämtliche Texte in Abbildungen werden in Englisch formuliert, um eine eindeutigere Benennung und Identifizierbarkeit auch im internationalen Kontext zu erreichen.

1.4. Motivation

Medizintechnik im allgemeinen und die Erfassung und Auswertung der EKG-Signale im besonderen sind sehr interessante Forschungs- und Entwicklungsfelder. Die Medizin ist in diesem Bereich sehr fortgeschritten, dennoch sind einige Erkrankungen des Herzens statistisch gesehen in den letzten Jahren in Deutschland gestiegen. Der Bedarf nach Geräten zur Aufzeichnung des EKG und somit zur rechtzeitigen Diagnose von Erkrankungen des Herzens ist hierbei ausschlaggebend.

Eine Übersicht zur Verdeutlichung dieses Sachverhaltes ist in Abbildung 1.1 dargestellt.

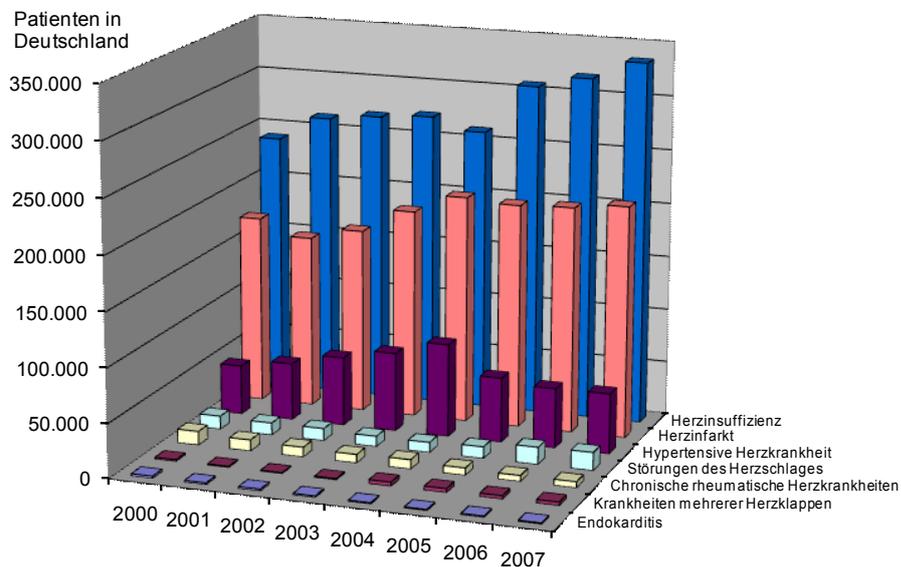


Abbildung 1.1.: Statistik von Herzerkrankungen in Deutschland

Die Statistik basiert auf Daten aus [Bundesamt 2009]. Im Diagramm sind mehrere Herzerkrankungen nach ihrem Auftreten und der Anzahl der betroffenen Patienten in Deutschland aufgeführt. Die Krankheiten mit den am stärksten betroffenen Patienten sind im hinteren Teil des Diagrammes angeordnet. Diese sind Herzinsuffizienz¹, Herzinfarkt, hypertensive Herzkrankheit² und Störungen des Herzschlages, auch als Herzrhythmusstörungen bekannt. Eine Ursache von Herzinsuffizienz können Herzrhythmusstörungen sein. Patienten, die an den oben genannten Krankheiten leiden bzw. gefährdet sind, müssen sich einem Langzeit-EKG unterziehen.

¹Krankhafte Unfähigkeit des Herzens, die benötigte Blutmenge ohne Druckanstieg in die Herzvorhöfe zu fördern.

²Durch den erhöhten Blutdruck, den die linke Herzkammer aufbringen muss, kommt es zu einer Verdickung der Muskulatur.

2. Systemkontext

Dieses Kapitel befasst sich mit der Einbettung des Monitors im Anwendungskontext.

Zuerst wird auf die Verteilung der Systemkomponenten eingegangen. Eine Diskussion der in Frage kommenden Nahfeldkommunikationsschnittstellen folgt darauf. Abschließend wird noch der Frage nachgegangen, inwieweit eine Fernfeldkommunikation zur Notfallmeldung realisierbar ist.

2.1. Verteilung

Der Einsatz des Monitors erstreckt sich auf die Erfassung des EKGs einer Person bzw. eines Patienten. Die dabei entstehenden Messdaten sollen vom Patienten selber und von einem Experten betrachtet und analysiert werden können. Dabei ist es wichtig, die Bewegungsfreiheit des Patienten nicht einzuschränken. Der Experte, der die Aufgabe hat eine Auswertung des EKGs vorzunehmen, ist meistens jedoch nicht vor Ort. Das UML-Verteilungsdiagramm 2.1 zeigt eine Möglichkeit der Realisierung, die Messdaten dennoch von den genannten Personen einsehen zu können.

Der EKG-Monitor enthält eine Monitor Unit, auf die in Kapitel 3 noch näher eingegangen wird. Diese Unit empfängt das EKG-Signal des Patienten³. Die Messwerte erreichen ein Gateway, die von dort aus per E-mail an den Patienten über das hausinterne Intranet oder an den Experten bzw. den Arzt des Patienten automatisch verschickt werden kann. Der Patient hat auf das Gateway über seinen PC Zugriff, um Einstellungen vornehmen zu können. Das Gateway dient hierbei auch als embedded Webserver⁴

³Das Signal wird über eine sogenannte Brustwandableitung abgegriffen. Eine geeignete Variante ist die Ableitung nach Nehb [EKGOnline 2009].

⁴Embedded Webserver sind Geräte, die in der Regel HTML-Seiten an den Webbrowser des Benutzers senden.

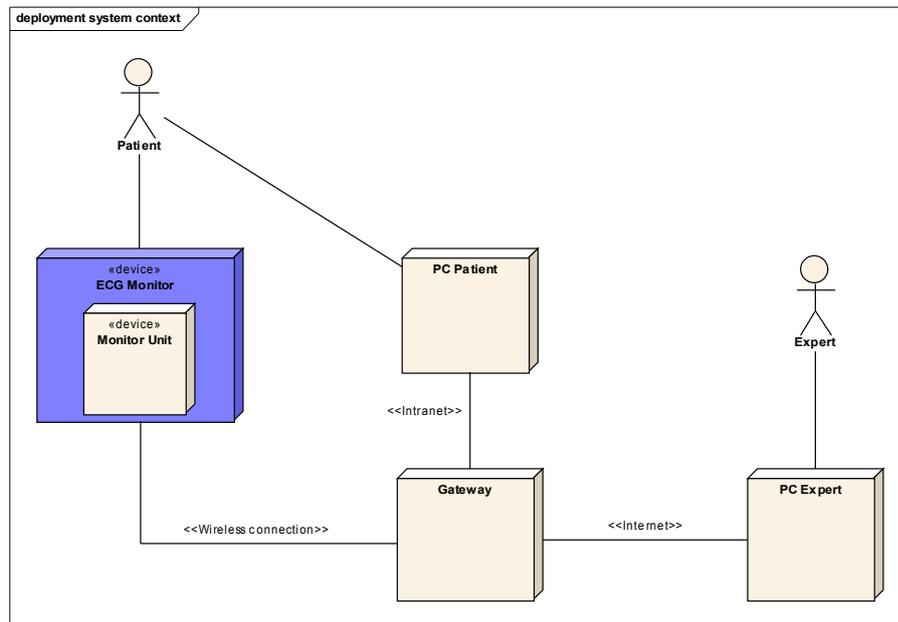


Abbildung 2.1.: Verteilung der Komponenten

2.2. Nahfeldkommunikation

Als Nahfeldkommunikation wird in diesem Kontext eine Entfernung von maximal 100 Meter zwischen Sender und Empfänger verstanden.

Für die Kommunikation zwischen Gateway bzw. Webserver des Patienten und dem Monitor in Abbildung 2.1 wird eine Funkverbindung geringerer Reichweite benötigt. Es sollen hierbei in den folgenden Kapiteln zwei mögliche Techniken begutachtet werden.

2.2.1. Bluetooth

IBM, Intel, Ericsson, Nokia und Toshiba gründeten 1998 eine Bluetooth Special Interest Group (SIG) zur Ausarbeitung eines Standards für den Ersatz von kabelgebundenen Datenverbindungen. Mit der Version 1.1 veröffentlichte die SIG 2001 einen Standard, der für die Entwicklung von marktgerechten Produkten als solide galt.

Die folgende Auflistung fasst die wichtigsten Eigenschaften von Bluetooth zusammen [Merkle 2002].

- Die Datenverbindung hat den Charakter einer None Light Of Sight (NLOS) Verbindung. Das bedeutet, es muss nicht Sichtkontakt zwischen Sender und Empfänger bestehen.

2. Systemkontext

- Bluetooth verwendet das 2,4GHz ISM Band, welches von WLAN ebenso benutzt wird.
- Um sich gegen Mithörern zu wehren, setzt man ein Frequenzspreizverfahren ein, auch als Frequency Hopping Spread Spectrum (FHSS) bekannt. Hierbei wechselt die Trägerfrequenz seinen Wert durch Pseudozufallszahlen.
- Die Bandbreite beträgt 80MHz, unterteilt in 79 Kanäle.
- Die Reichweite wird in 3 Klassen unterteilt:

Tabelle 2.1.: Bluetooth-Reichweiten in Abhängigkeit der Klassen [Merkle 2002]

Klasse	Leistung	Reichweite
1	100mW (20dBm)	100m
2	2,5mW (4dBm)	ca. 20m
3	1mW (0dBm)	10m

- Bluetooth kennt bis Version 2.0 die Fehlerbehandlung FEC (Forward Error Correction) und ARQ (Automatic Repeat Request)
- Mit Bluetooth können Ad-Hoc Verbindungen spontan und autonom aufgebaut werden. Benutzt werden hierbei Piconet und Scatternet⁵.

Kernstück von Bluetooth ist die Protokollarchitektur. Diese ist in Abbildung 2.2 dargestellt.

Die Protokollarchitektur besteht aus einem Bluetooth Chip (Funktionsblöcke unterhalb vom Host-Controller-Interface dargestellt) und einem Bluetooth Stack (oberhalb davon). Der Bluetooth Stack ist ein Softwarepaket mit Treibern, welches von verschiedenen Herstellern bezogen werden kann. Der Stack muss sowohl im Sender als auch im Empfänger implementiert sein.

Um eine serielle Datenverbindung nutzen zu können, muss der Stack das Profil SPP (Serial Port Profile) unterstützen. Dieses Profil ist mit der RS232-Schnittstelle vergleichbar. Die Daten gelangen nun über RFCOMM, L2CAP, dem Baseband Chip (Bluetooth Chip) zur Luftschnittstelle (Radio). RFCOMM steht hierfür für Radio Frequency Communication und L2CAP für Logical Link Control and Adaption Protocol.

Der Block SDP (Service Discoverability Protocol) wird benötigt, um der Gegenstelle mitzuteilen, welche Services (z.B. SPP) unterstützt werden. Für die Telefonie zum Beispiel

⁵Piconet ist eine Ansammlung von maximal 7 slaves und 1 master. Scatternet besteht aus 2 oder mehr Piconets

2. Systemkontext

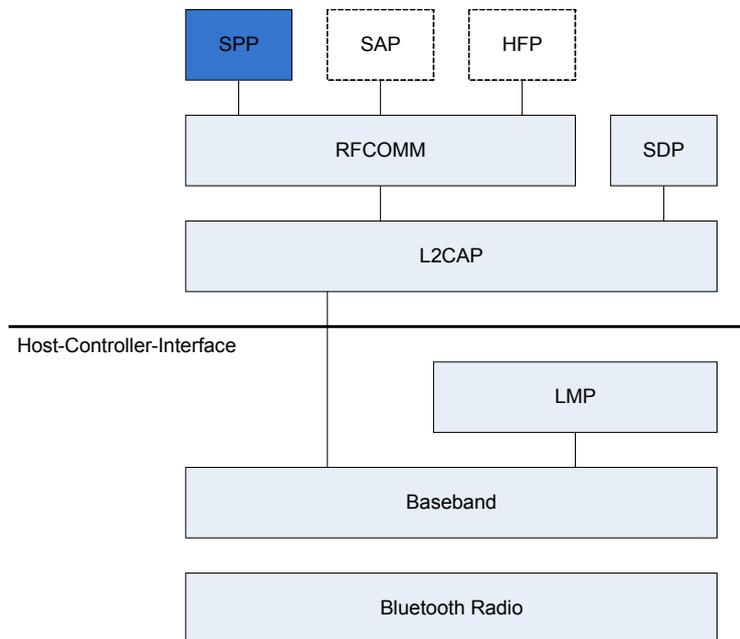


Abbildung 2.2.: Bluetooth Protokoll Stack (Bildquelle: [Merkle 2002] modifiziert)

gibt es die Profile SIMAP und HFP (SIM Access Profile und Hands-Free Profile)⁶. Der Block LMP (Link Manager Protocol) ist zuständig für den grundlegenden Verbindungsaufbau über die Luftschnittstelle.

2.2.2. Simple Serial Wireless Connection

Simple Serial Wireless Connection, im folgenden als SSWC abgekürzt, ist keine geschützte Bezeichnung wie Bluetooth, sondern steht für eine einfache Realisierung einer seriellen Datenverbindung. Wie der Name schon sagt handelt es sich hierbei um eine Datenübertragung, hinter der sich eine schlanke Architektur verbirgt. Diese Architektur zeigt Abbildung 2.3.

Der unterste Block, SSWC Radio, steht für die Funkausbreitung in der Luft. Im einfachsten Fall handelt es sich hierbei um FSK (Frequency Shift Keying). Das FM-Modul setzt dafür den leitungscodeierten Bitstrom in das entsprechende trägermodulierte Signal um. Durch das Single Wire Interface wird dieser Bitstrom übertragen. Zuständig für die Erzeugung des Signals ist der Block BPMC. Dieser steht für BiPhase Mark Coding. Die zugrunde liegende Leitungscodierung nennt sich Biphas-Mark [BiPhaseMark 2009]. Hierbei findet an jedem Bit-Intervall-Anfang ein Pegelwechsel statt, während in der Bit-Intervall-Mitte ein Wechsel

⁶Mit SIMAP kann eine in einem Mobile befindliche SIM Karte über Bluetooth zugänglich gemacht werden. Viele Freisprechgeräte in Fahrzeugen benutzen HFP zur Telefonie.

2. Systemkontext

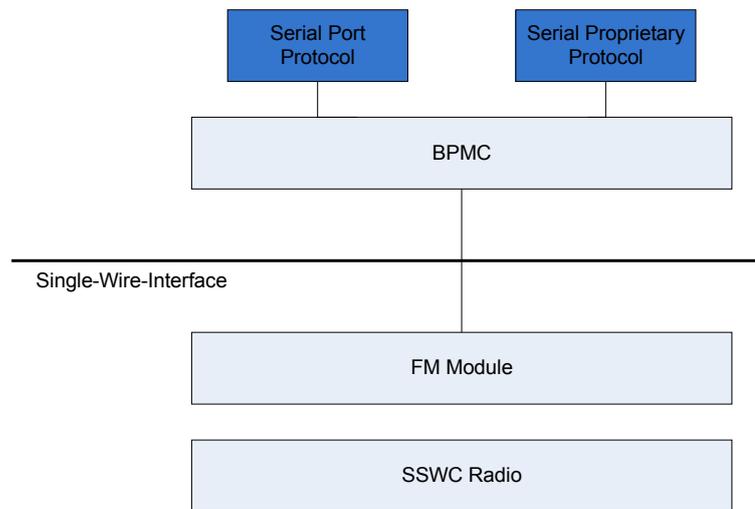


Abbildung 2.3.: Simple Serial Wireless Connection Schichtenmodell

nur bei 1-Bits stattfindet. Mit diesem Code ist es möglich, den Grundtakt des Bitstroms beim Empfänger zurück zu gewinnen. Alle BiPhase-Codes weisen nur geringe niederfrequente Spektralanteile auf. Darüber hinaus sind sie einfach zu implementieren. Einsatz finden sie u.a. im LAN-Bereich und in MOST⁷ Bussen. BPMC dient sozusagen als Gateway zwischen einem Serial Port Protocol (angelehnt wie bei Bluetooth an RS232), oder auch eventuell einem proprietären Protokoll. Eine Implementierung für BPMC-Empfänger und BPMC-Sender ist auf der CD zur Masterarbeit unter */C-Algorithm/BiPhaseMark* enthalten. Ein Empfänger, bei welchem dies umgesetzt wurde, zeigt das Foto in Abbildung 2.4.

Im folgenden werden ein paar technische Daten zu diesem Empfänger zusammengestellt:

- Empfangsfrequenz: 433,92MHz
- Empfindlichkeit: -111dBm
- Bandbreite: 600kHz

⁷Media Oriented Systems Transport, hierbei handelt es sich um ein serielles Bussystem in Ringtopologie zur Übertragung von Multimedia-Daten

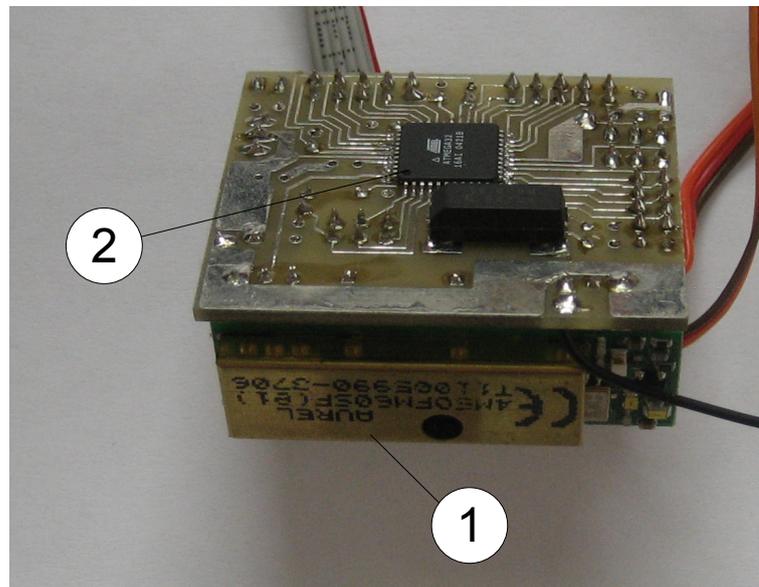


Abbildung 2.4.: Empfänger zur Datenübertragung mit SSWC, 1) FM-Modul, 2) Prozessor für BPMC

2.3. Fernfeldkommunikation

Dieses Kapitel ist der Überlegung gewidmet, eine Art Notrufmeldung zu diskutieren. Die Notrufmeldung ermöglicht das Alarmieren eines Notdienstes im Falle eines gesundheitgefährdeten Zustandes des Patienten. Solch ein Zustand könnte z.B. ein Herzinfarkt sein. Voraussetzung dafür ist, dass dieser akute Zustand vom EKG-Monitor in Abbildung 2.1 erkannt werden kann. Für die Alarmierung kann das GSM-Netz hergenommen werden. Hierfür existieren alle notwendigen Einrichtungen. Eine Einrichtung, an die sich Patienten oder Unfallopfer richten können, ist neben den bekannten Notfall-Einrichtungen die Björn-Steiger-Stiftung [Steiger 2009]. Im Mittelpunkt stehen eigene Aktivitäten sowie die Aktivierung staatlicher und privater Institutionen zur Verbesserung der Notfallhilfe.

In der Automobilindustrie gibt es bereits Entwicklungen von Geräten, die bei einem Unfall selbstständig Notfallhilfe anfordern. Übertragen auf den EKG-Monitor könnte dies auch eingesetzt werden. In Abbildung 2.5 ist solch ein System schematisch dargestellt.

Die Monitor Unit wird dazu um ein GSM-Modul, SIM-Kartenleser und GPS-Modul erweitert. Das GPS-Modul hat die Aufgabe, die Position des Patienten zu erfassen, falls er sich außer Haus befindet. Das GSM-Modul kann als eine Telefoneinheit verstanden werden, mit der man telefonieren und die SMS verschicken kann. Ohne SIM-Karte kann sich das GSM-Modul nicht in das Netz einwählen. Bisher war es möglich, ohne SIM-Karte Notrufe

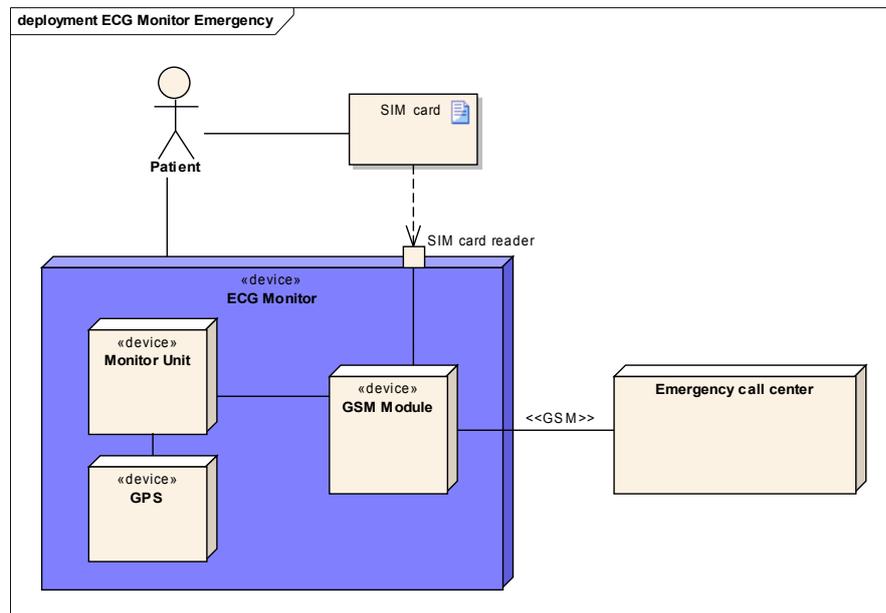


Abbildung 2.5.: Notruffunktionalität

absetzen zu können, dies wurde aber per Gesetz nun ab 1. Juli 2009 verboten. Als Notfallszenario kann man sich einen Patienten mit Herzinfarkt vorstellen. Die Feststellung des Infarktes übernimmt die Monitor Unit. Sind GPS-Daten verfügbar, kann die Position des Patienten in einer SMS an die Björn-Steiger-Stiftung versandt werden. Parallel dazu wird eine Notrufnummer abgesetzt. Stehen keine GPS-Daten zur Verfügung kann eventuell auch auf eine Variante zurückgegriffen werden, bei der einfach eine SMS versandt wird mit Name und Adresse des Patienten.

2.4. Resümee

Im Rahmen dieser Arbeit kann neben der Fernfeldkommunikation nur auf das Thema der Nahfeldkommunikation weiter eingegangen werden. Zur Gegenüberstellung der oben erwähnten Nahfeldkommunikationstechniken Bluetooth und SSWC folgt die Tabelle 2.2.

Tabelle 2.2.: Gegenüberstellung von Bluetooth und SSWC

Gegenüberstellungspunkt	Bluetooth	SSWC
Lizenzkosten	Ja, für kommerziellen Bluetooth Stack ⁸	Nein
Verfügbarkeit von Modulen	Ja	Ja
Implementierungsaufwand	hoch	niedrig
Netztopologie	point to multipoint	point to point
Erweiterbarkeit	möglich, da verschiedene Profile verfügbar	nicht möglich
Fehlerbehandlung	FEC und ARQ	FEC

Beim Einsatz von Bluetooth für kommerzielle Zwecke können Lizenzgebühren anfallen. Für beide Arten existieren Module, die in HW-Schaltungen eingebaut werden können. Der Implementierungsaufwand ist bei Bluetooth wesentlich höher, da die Architektur dazu komplexer ist. Dagegen unterstützt BT Point-to-Multipoint-Verbindungen. Bezüglich den BT-Diensten unterstützt BT neben SIMAP, HFP und SPP noch weitere. Der Vorteil dabei ist, dass neue Dienste einfach durch SW-Erweiterungen umgesetzt werden können. Die beschriebenen Vorteile bringen für den EKG-Monitor keinen Mehrwert. Im weiteren Verlauf wird deshalb Simple Serial Wireless Connection (SSWC) weiter verfolgt.

⁸Es gibt open source Implementierungen wie z.B. IwBT, generell sind sie aber abhängig vom Prozessor und dem einzusetzenden Profil.

3. Aufbau

Dieses Kapitel beschäftigt sich mit dem Aufbau der zentralen Bestandteile des EKG-Monitors. Darüber hinaus beinhaltet dieses Kapitel weitere Festlegungen zur Konzeption. Die Bestandteile werden anhand der Signalverarbeitungskette dargestellt und im Kontext der Monitor Unit beschrieben. Es folgt eine Einführung über den Aufbau der Signalquelle. Es wird auf die Notwendigkeit einer Signalaufbereitung hingedeutet. Die grundlegenden Quellen- und Kanalcodierverfahren werden dargestellt und die geeigneten Methoden für die EKG-Anwendung herausgearbeitet. Abschließend folgt ein Domain-Modell zur Zusammenfassung der ausgewählten Techniken.

3.1. Übersicht

Bevor ein Signal gespeichert oder übertragen wird, durchläuft es eine Verarbeitungskette. Abbildung 3.1 zeigt ein Übertragungsmodell zu diesem Zweck.

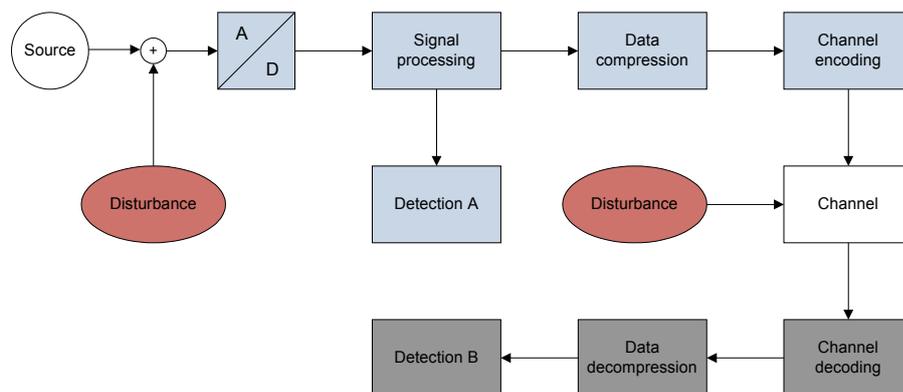


Abbildung 3.1.: Modell der Verarbeitungskette (Bildquelle: [Strutz 2005] modifiziert)

Die hellblauen Funktionsblöcke im oberen Bereich sind Bestandteil des mobilen EKG-Monitors (Monitor Unit), wohinhingegen die unteren grauen Blöcke Bestandteil des Gateways bzw. des Webservers sind.

Die Signalquelle (Source) ist mit Störungen (Disturbance) beaufschlagt. Nachdem das Signal durch einen Analog-/Digital-Wandler digitalisiert wurde, sorgt die Signalaufbereitung

3. Aufbau

(Signalprocessing) zur Störbeseitigung und Vorbereitung für die weitere Verarbeitung. Die Detektion A (Detection A) des EKG-Monitors erkennt das EKG-Signal und dient zur Vorbereitung der Notfallerkennung (z.B. Herzinfarkt). Die von der Signalquelle erzeugten Daten werden durch die Datenkompression (Data compression) komprimiert, damit sie effizient übertragen werden können. Die nachfolgende Kanalcodierung (Channelencoding) fügt gezielt wieder etwas Redundanz hinzu, um einen Fehlerschutz oder zumindest eine Fehlererkennung zu ermöglichen. Durch die Kanalcodierung wird der Signalfluss an die Eigenschaft des Kanals (Channel) angepasst⁹. Zur Unterscheidung der Verarbeitung auf Sender- und Empfängerseite wird hier zwischen Encodierung und Decodierung unterschieden. Im stationären Teil bzw. im Gateway müssen die Daten wieder decodiert (Channel decoding) und teilweise dekomprimiert (Data decompression) werden, um das EKG-Signal wieder soweit zurück zu gewinnen, dass eine Detektion (Detection B) möglich ist. Die Detektion B ist somit mit der Detektion A im mobilen Teil des Monitors vergleichbar.

Überträgt man die Verarbeitungskette auf den EKG-Monitor, so bilden die hellblau gekennzeichneten Funktionsblöcke die Monitor Unit in Abbildung 3.2.

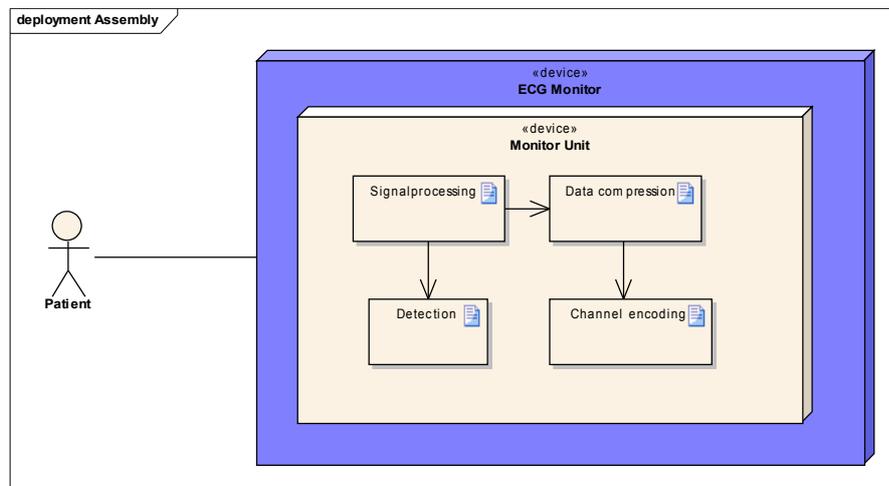


Abbildung 3.2.: Übersicht der in der Monitor Unit eingesetzten Funktionsblöcke

Für die Zwischenspeicherung der Daten auf der Monitor Unit hat die Datenkompression den Vorteil, dass hierbei die Daten komprimiert abgelegt und somit Speicherplatz gespart werden kann.

⁹Jeder Kanal unterliegt Beschränkungen und Störungen (Disturbance) wie z.B. Bandbreite und Fehler-rate. Zur Erkennung der Störung können beispielsweise Paritäts-Bits oder Checksummen eingesetzt werden.

3.2. Signalquelle

Um das Signal elektrisch erfassen zu können, muss es in ein elektrisches Signal umgewandelt werden. Dazu müssen Elektroden am Körper des Patienten angebracht werden¹⁰. Man unterscheidet zwischen Extremitäten- und Brustwandableitungen¹¹. Die bekannteste Extremitätenableitung ist die Einthoven-Ableitung, dargestellt in Abbildung 3.3.

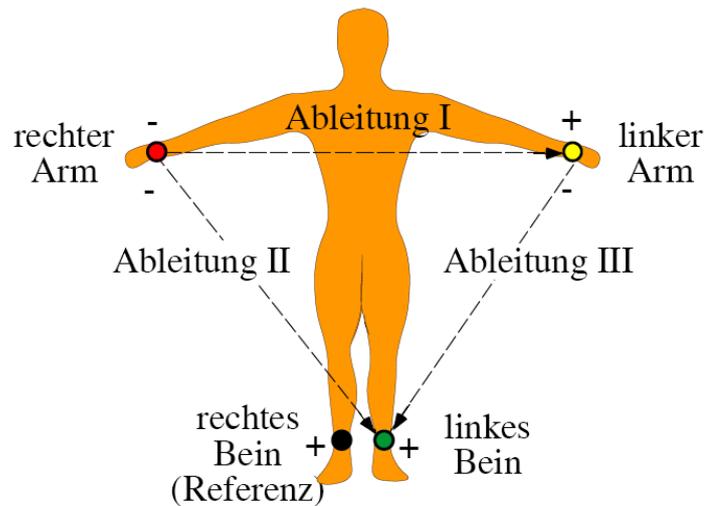


Abbildung 3.3.: Einthoven-Ableitungen (Bildquelle: [IfBTUK 2009])

[So 2001] beschreibt diese Ableitungen als die allerwichtigsten und grundlegendsten Ableitungen in der Elektrokardiographie. Diese teilen sich in Ableitung I, II und III auf¹². Die Einthoven-Ableitungen bilden zusammen das sogenannte Einthoven-Dreieck. Diese entstehen durch die Verbindung von jeweils zwei Extremitätenableitungen („bipolare“ Ableitungen). Die Elektroden sind dabei folgendermassen verbunden:

- Ableitung I: **rechter Arm (rot)** und **linker Arm (gelb)**
- Ableitung II: **rechter Arm (rot)** und **linkes Bein (grün)**
- Ableitung III: **linker Arm (gelb)** und **linkes Bein (grün)**

Die Extremitätenableitungen sind aber zur Patientenüberwachung im Heimbereich untauglich, da die Bewegungsfreiheit nicht gewährleistet ist. Hierfür sind die Brustwandableitungen besser geeignet, da die dafür notwendigen Elektroden im Brustbereich angebracht

¹⁰Zwischen Haut und Elektrode wird mit physiologischer NaCl-Lösung getränktes Elektrodenpapier gelegt, um den Übergangswiderstand Haut-Elektrode zu verringern.

¹¹Das „Abgreifen“ des Signals vom Körper wird als Ableitung bezeichnet.

¹²Heute noch wird die Ableitung II zur Diagnose von Herzrhythmusstörungen eingesetzt.

3. Aufbau

werden. Dabei überträgt man das Herzdreieck nach Einthoven auf den Brustbereich. Es entsteht das kleine Herzdreieck nach Nehb in Abbildung 3.4.

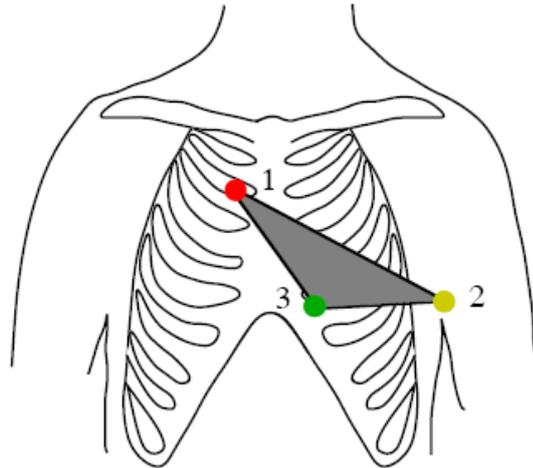


Abbildung 3.4.: Nehb Ableitung (Bildquelle: [IfBTUK 2009])

Für die Ableitungen gelten folgende Bezeichnungen:

- Dorsal: **Elektrode 1 (rot)** und **Elektrode 2 (gelb)**
- Anterior: **Elektrode 1 (rot)** und **Elektrode 3 (grün)**
- Inferior: **Elektrode 2 (gelb)** und **Elektrode 3 (grün)**

Das Fraunhofer-Institut [FIPM 2009] bietet für die Brustwandableitung eine EKG-Folie in Abbildung 3.5 an. Diese Folie hat den Vorteil, dass die Elektroden leichter angebracht werden können.

In Abbildung 3.1 ist neben der Signalquelle (Source) noch eine weitere Quelle, als Disturbance (Störquelle) bezeichnet, zu sehen. Die Störquelle ist der Signalquelle überlagert. Nach [Krishnan 2007] kommen als Ursache folgende Störquellen in Frage:

- Bewegungsartefakte der Ableitelektroden¹³.
- Der Vorverstärker verursacht aufgrund seines grossen Verstärkungsfaktors Messrauschen.
- Störanteile aufgrund der Netzspannung mit einer Frequenz von 50Hz.
- Höherfrequente Einkopplungen (>50Hz) aufgrund von EMV (Elektromagnetische Verträglichkeit) anderer Geräte wie Mobiltelefone etc.

¹³Jede Bewegung an der Kontaktstelle ändert die Kontaktspannung. Ursache hierfür ist u.a. das Atmen.

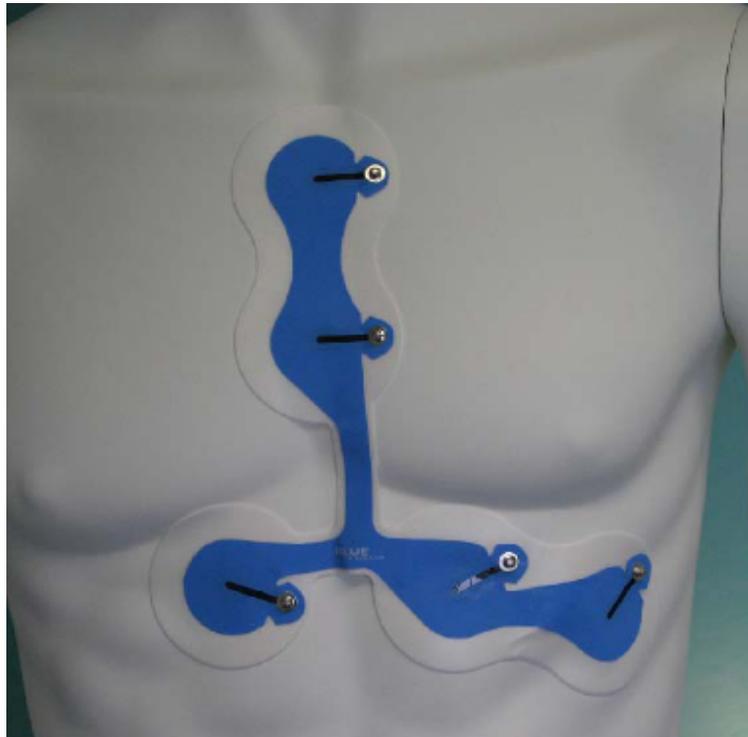


Abbildung 3.5.: EKG Folie (Bildquelle: [FIPM 2009])

3.3. Signalaufbereitung

Die Signalaufbereitung bzw. Signalprocessing laut Abbildung 3.1 hat die Aufgabe, das ursprüngliche EKG Signal von den Störanteilen weitgehend zu befreien. Die relevanten Störanteile wurden bereits in obiger Auflistung zusammengefasst. Darüber hinaus muss das Signal für die Datenkompression (Data compression) vorbereitet werden.

Anwendung zur Signalaufbereitung finden hauptsächlich digitale Filter. In [Grueningen 2002] werden diese durch ihre Übertragungsfunktion $H(z)$ in der Z-Ebene folgendermassen charakterisiert:

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + \dots + a_Mz^{-M}} \quad (3.1)$$

Sind alle Koeffizienten a_i gleich Null, dann spricht man von einem nicht rekursivem Filter bzw. einem FIR-Filter¹⁴. Von einem rekursiven Filter bzw. IIR-Filter spricht man, wenn die Koeffizienten b_i gleich Null sind. Im Zeitbereich lautet die zugehörige Differenzgleichung $y[n]$:

¹⁴FIR - Finite Impulse Response, IIR - Infinite Impulse Response.

3. Aufbau

$$y[n] = \sum_{i=0}^N b_i x[n-i] - \sum_{i=1}^M a_i y[n-i] \quad (3.2)$$

In Kapitel 5 werden die für die Signalaufbereitung notwendigen Filter genauer spezifiziert.

3.4. Datenkompression

Grundsätzlich unterteilt man die elementaren Verfahren zur Datenkompression in drei Gruppen [Strutz 2005]. Die erste Gruppe wird mit dem Begriff Datenreduktion beschrieben. Dazu gehören alle Verfahren, die Daten irreversibel aus dem Signal entfernen und somit das Signal *verlustbehaftet* verändern. Die zweite Gruppe umfasst alle Codierungsverfahren, welche *verlustlos* die Daten verarbeiten. Ergänzt werden die zwei Gruppen durch die Dekorrelation. Ziel der Dekorrelation ist es, eine Konzentration der Information herbeizuführen. Tabelle 3.1 fasst die drei elementaren Verfahren zusammen.

Tabelle 3.1.: Einteilung von Techniken zur Verringerung der Datenmenge

Datenkompression		
Datenreduktion	Codierung	Dekorrelation
Entfernen von Irrelevanz	Entfernen von Redundanz	Konzentration der Signalenergie bzw. der Information

Die Datenreduktion kennt im Prinzip zwei Verfahren zur Verringerung der Datenrate, die Quantisierung und die Unterabtastung (Tabelle 3.2).

Bei der Unterabtastung wird nur jeder M-te Signalwert hergenommen:

$$y[m] = x[m \cdot M] \quad (3.3)$$

Durch die Unterabtastung wird das Spektrum des Signals beeinflusst. Die Abtastrate f_s verringert sich um den Faktor $1/M$.

$$f_{sn} = \frac{f_s}{M} \quad (3.4)$$

3. Aufbau

Ein entsprechendes Antialiasing-Filter muss vor der Unterabtastung dafür sorgen, dass sich die Frequenzbänder nach der Unterabtastung nicht überschneiden¹⁵.

Tabelle 3.2.: Unterteilung der Verfahren zur Datenreduktion

Datenreduktion	
Quantisierung	Unterabtastung
Verringerung der Auflösung der Signalwerte	Verringerung der zeitlichen bzw. örtlichen Auflösung

Bei der Quantisierung wird der Messbereich der analogen Größe in eine endliche Zahl aneinander angrenzende Teilbereiche (Intervalle) aufgeteilt und jedem davon ein Wert eines endlichen symbolischen Systems zugeordnet (z. B. eine ganze Zahl). Um eine effektive Datenreduktion zu erreichen, wird eine Quantisierung von 8Bit angestrebt.

Wie bereits erwähnt ist das Ziel der Dekorrelation in Tabelle 3.3, Information zu konzentrieren. Je stärker die Information bzw. die Signalamplituden um einen Wert konzentriert sind, desto kleiner ist die Signalentropie. Bei einer nachfolgenden Codierung kann dann den häufig auftretenden Zeichen ein kürzerer Code zugewiesen werden. Dekorrelation kann mit Prädiktion erzielt werden. Auf das Filterbank- und Transformation-Verfahren soll auf die Literatur [Strutz 2005] verwiesen werden.

Tabelle 3.3.: Verfahren zur Dekorrelation von Signaldaten

Dekorrelation		
Prädiktion	Filterbank	Transformation
Voraussage von Signalwerten	Zerlegen von Signalen in (überlappende) Frequenzbereiche	Zerlegen von Signalen in Basisfunktionen

Die Codierung in Tabelle 3.4 teilt sich in zwei Bereiche auf: Die Präcodierung und die Entropiecodierung. Aufgrund des Zusammenhangs der Prädiktion mit der damit verbundenen Dekorrelation bietet sich für die Codierung die Entropiecodierung an. Die Entropie ist definiert durch:

$$H(x) = - \sum_{i=1}^q p(x_i) \cdot \text{ld}(p(x_i)) \quad (3.5)$$

¹⁵Überschneiden sich die Frequenzbänder treten unerwünschte Frequenzanteile in dem interessierendem Frequenzband auf.

3. Aufbau

Die Einheit ist [bit/Zeichen]. Die Entropie erhält ihr Maximum bei einer Gleichverteilung der Daten. Je kleiner die Entropie, desto weniger Anzahl Bits muss für ein Zeichen aufgebracht werden.

Tabelle 3.4.: Unterteilung der Verfahren zur Codierung

Codierung	
Präcodierung	Entropiecodierung
Betrachten der Symbole als abhängig voneinander	Betrachten der Symbole als unabhängig voneinander
Abbilden von Symbolen auf Symbole eines anderen Alphabets oder Aufspüren von Korrelationen	Abbilden der Symbole auf Codewörter
Ziel: Minimieren der Intersymbolredundanz	Ziel: Annähern des Speicheraufwandes an die Signalentropie
Algorithmus: Lauflängencodierung, Phrasencodierung	Algorithmus: Huffman-Codierung, arithmetische Codierung

Eine Methode zur Generierung der Codes für die Entropiecodierung ist die Huffman-Codierung. Sie generiert Präfixcodes¹⁶ und garantiert eine optimale Codewortzuweisung.

3.5. Kanalcodierung

Die Kanalcodierung (Channelencoding) aus Abbildung 3.1 fügt den Daten am Eingang eines Übertragungskanals Redundanz hinzu. Wenn die Zusatzinformationen lediglich auf einen Fehler hindeuten und eine Neuübertragung der Daten erforderlich machen, spricht man von Wiederholungsanforderung, abgekürzt ARQ (Automatic Repeat Request). Genügt die Redundanzinformation, den Fehler zu korrigieren, handelt es sich um eine Vorwärtsfehlerkorrektur (FEC - Forward Error Correction).

Tabelle 3.5 zeigt eine Übersicht der wichtigsten Codeklassen.

¹⁶Diese Codes sind präfixfrei, d.h. die Anfangszeichenfolge des Codewortes kommt in keinem anderen Codewort vor

3. Aufbau

Tabelle 3.5.: Codeklassen der Kanalcodierung

Codeklassen				
lineare Blockcodes			Faltungscodes	
Zyklische Blockcodes	Hamming-Codes	Einfache Paritäts- und Wiederholungs-codes	Nicht rekursive nicht systematische Codes (NSC-Codes)	Rekursive systematische Codes (RSC-Codes)

Auf die Faltungscodes wird nicht näher eingegangen, da der zur Decodierung notwendige Algorithmus relativ umfangreich in der Implementierung ist¹⁷.

Bei den linearen Blockcodes gibt es eine einfache Realisierung der Decodierung durch *Syndromdecodierung* [Carl 2009].

Zur einfachen Beschreibung der Blockcodes wird folgende Form eingeführt:

$$(m, n, d_{min}) - \text{Blockcode} \tag{3.6}$$

Dabei steht m für die Anzahl Stellen des Codewortes und n für die Anzahl Stellen des Informationswortes. Die Hammingdistanz wird mit d_{min} bezeichnet. Hamming-Codes beispielsweise besitzen eine Hammingdistanz von $d_{min} = 3$ und erlauben somit eine Fehlererkennung von 2 Einzelfehlern. Beim einfachen Paritätscode beträgt die Distanz nur $d_{min} = 2$. Die zyklischen Blockcodes lassen sich noch weiter unterteilen:

Zyklische Blockcodes:

- CRC-Codes (CRC = Cyclic Redundancy Check)
- Zyklische Hammingcodes
- Fire-Codes
- BCH-Codes (BCH = Bose-Chaudhuri-Hocquenghem)
- Reed-Solomon-Codes

An dieser Stelle soll der CRC-Code hervorgehoben werden. Auf die restlichen Codes wird auf die Literatur [Bossert 1998] verwiesen. Der Grund dafür besteht in der Fehlererkennungsfähigkeit von CRC-Codes. Diese lässt eine *Bündelfehlererkennung* zu. Gegenüber

¹⁷Für die Faltungscodes wird häufig der Viterbi Algorithmus [Bossert 1998] hergenommen.

3. Aufbau

dem Rauschen, bei dem hin- und wieder Einzelfehler auftreten, treten bei Bündelfehlern¹⁸ die Einzelfehler gehäuft auf.

3.6. Detektion

Nachdem das Signal aufbereitet bzw. übertragen und decodiert wurde, erfolgt die Detektion des EKG-Signals (Detection B in Abbildung 3.1). Das Signal ist dann soweit von den Störungen (Disturbance) befreit bzw. unterdrückt, dass eine Detektion erfolgen kann. Die Detektion ermöglicht eine optische Auswertung des Signals. Unter optischer Auswertung wird eine Bewertung des Amplitudenverlaufs im Zeitbereich verstanden. Die Auswertung soll es erlauben, Herzrhythmusstörungen, Herzinfarkte und sonstige Auffälligkeiten des EKG-Signals hervorzuheben [So 2001].

3.7. Domain-Modell

Als Zusammenfassung der in diesem Kapitel vorgestellten Techniken dient das Domain-Modell in Abbildung 3.6. Die darin vorgestellten Techniken werden in den folgenden Kapiteln weiter erläutert und beschrieben.

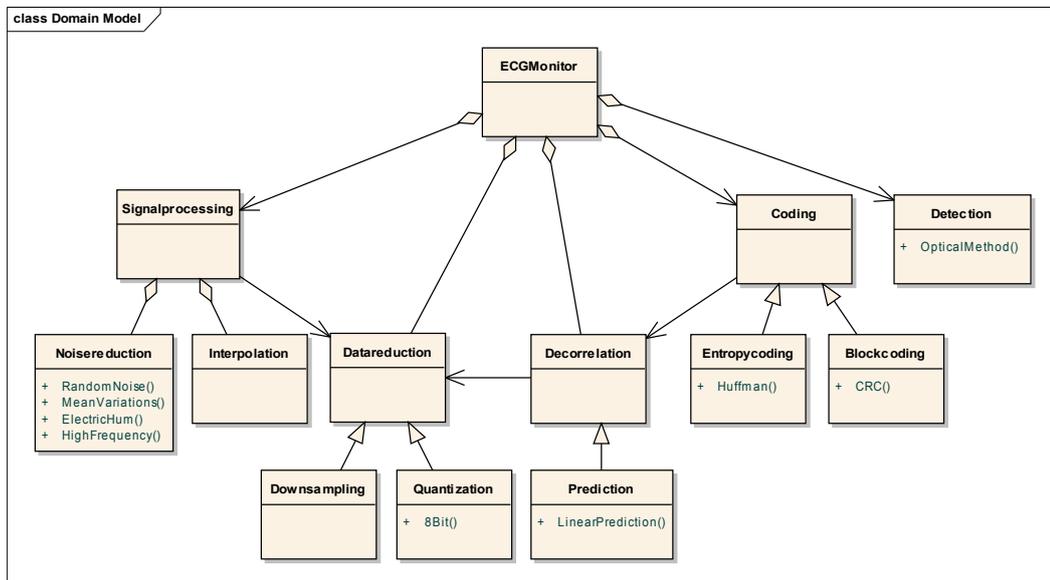


Abbildung 3.6.: Domain-Modell für die verwendeten Techniken

¹⁸Ursache hierfür sind elektromagnetische Bursts, wie z.B. durch das Einschalten eines Relais bedingte Auftreten von Störfeldern.

3. Aufbau

Der EKG-Monitor besteht somit aus drei Hauptkomponenten: Der Signalverarbeitung (Signalprocessing), der Codierung (Coding) und der Detektion (Detection). Die Signalverarbeitung wiederum besteht aus der Störunterdrückung (Noisereduction) und der Signalinterpolation¹⁹ (Interpolation). Zu den in Kapitel 3.2 aufgeführten Störquellen werden die folgenden vier Methoden zur Störunterdrückung benötigt:

- Die Methode *MeanVariations()* zur Unterdrückung von Mittelwertschwankungen.
- Die Methode *RandomNoise()* zur Unterdrückung von Messrauschen.
- Die Methode *ElectricHum()* zur Unterdrückung von Netzbrummen.
- Die Methode *HighFrequency()* zur Unterdrückung von höherfrequenten Störeinkopplungen.

Die Signalverarbeitung benutzt dazu mitunter Methoden aus der Datenreduktion. Dies sind Downsampling und Quantisierung (Quantization), um die Signalwerte zu komprimieren und zu digitalisieren. Wie bereits erwähnt wird eine Quantisierung von 8Bit angestrebt. Die Dekorrelation wird mittels linearer Prädiktion umgesetzt und benötigt ebenso die Quantisierung. Die Codiervorgänge beschränken sich auf die Huffman-Codierung sowie auf die linearen Blockcodes *CRC()* zur Kanalcodierung. Bei der Detektion (Detection) kommt das „optische“ Verfahren *OpticalMethod()* zum Einsatz.

¹⁹Die Interpolation beinhaltet das Upsampling (Aufwärtstastung) und die Interpolationsfilterung um die Zwischenpunkte der eingefügten Werte zu berechnen.

4. Elektrokardiogramm

Dieses Kapitel erklärt die Herzfunktion und die damit verbundene Entstehung des EKG-Signals. Das Signal wird dann in einzelne Abschnitte eingeteilt und beschrieben. Darauf folgt eine erste Abschätzung des EKG-Frequenzspektrums für die spätere Filterspezifikation. Abschließend wird eine EKG-Simulation zur weiteren Verwendung mit Matlab vorgestellt.

4.1. Entstehung

In Abbildung 4.1 wird mit den vier Schemata Zeichnungen die prinzipielle Funktionsweise und Entstehung des EKG-Signals veranschaulicht [So 2001].

Im Bild links oben befindet sich das Herz im Anfangszustand kurz vor dem ersten Erregungszustand. Der Sinusknoten ist das eigentliche Schrittmacherzentrum. Der Reiz wird primär hier gebildet. Zu Beginn erregt der Sinusknoten die Kammervorhöfe²⁰. Die Vorhofmuskulatur kontrahiert²¹ und drückt das Blut durch die Mitralklappe und Trikuspidalklappe in die Herzkammern²² (Abbildung 4.1 Bild rechts oben). Der AV-Knoten sorgt nun für die Kammererregung, die Herzkammermuskulatur kontrahiert und das Blut fließt durch die Aortenklappe und Pulmonalklappe²³ ab. Im Bild rechts unten tritt nun eine Erregungsrückbildung ein, um die Herzmuskulatur in den Anfangszustand zurück zu versetzen. Das Blut fängt wieder an, durch die Vorkammerhöfe zur Herzkammer zu fließen und der Kreislauf beginnt von vorn.

²⁰Die Kammervorhöfe liegen oberhalb Mitralklappe und Trikuspidalklappe.

²¹In der medizinischen Literatur ist sowohl von kontrahieren als auch von kontrahieren die Rede.

²²Die Herzkammern befinden sich unterhalb Mitralklappe und Trikuspidalklappe.

²³Die Pulmonalklappe befindet sich in Abbildung 4.1 hinter der Aortenklappe und ist somit nicht zu sehen.

4. Elektrokardiogramm

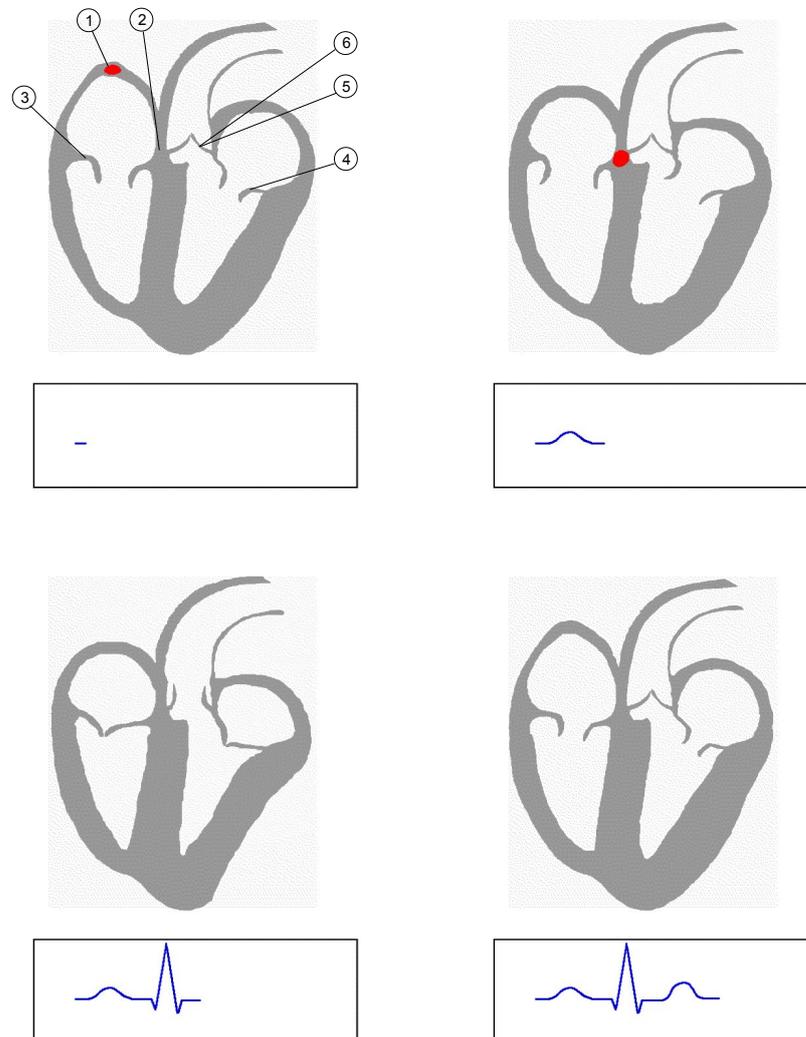


Abbildung 4.1.: Entstehung des EKG-Signals (Bildquelle: [EKGEntstehung 2009] modifiziert), 1) Sinusknoten, 2) AV-Knoten, 3) Mitralklappe, 4) Trikuspidalklappe, 5) Aortenklappe und dahinter Pulmonalklappe

4.2. EKG-Abschnitte

Die EKG-Abschnitte bezeichnen die einzelnen Phasen der Herzerregungszustände. Aus der Herzfunktion (Abbildung 4.1) leiten sich die elektrischen Pulse ab. Man fasst die Abschnitte in 3 Phasen zusammen, wie in Abbildung 4.2 dargestellt:

- Erregung der Kammervorhöfe
- Die Herzkammererregung
- Die Erregungsrückbildung

4. Elektrokardiogramm

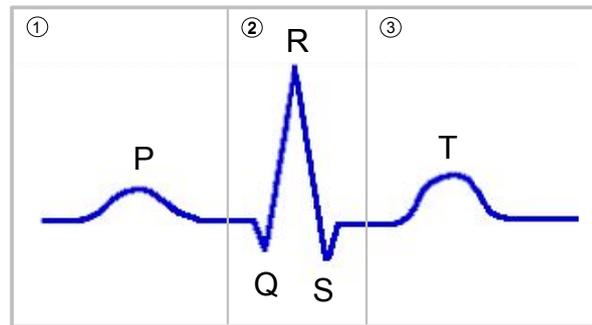


Abbildung 4.2.: Abschnitte des EKG-Signals, 1) Erregung der Vorhöfe, 2) Kammererregung, 3) Erregungsrückbildung

Zur eindeutigen Identifizierbarkeit der Pulse, auch Zacken genannt, werden diese mit den Buchstaben P,Q,R,S,T bezeichnet. Aus den Erfahrungen der Elektrokardiographie haben sich für die Pulse Normalwerte ergeben, die in Tabelle 4.1 zusammengefasst sind [So 2001].

Tabelle 4.1.: Normalwerte für Amplitude und Dauer der EKG-Pulse

Puls	Amplitude	Dauer
P	$\leq 0,20\text{mV}$	$\leq 0,11\text{s}$
Q	$\frac{1}{4}$ von R	$\leq 0,04\text{s}$
R	$0,6 - 2,6\text{mV}$	$\leq 0,05\text{s}$
S	k.A.	$< 0,06\text{s}$
T	$\frac{1}{7}$ von R	$0,19 - 0,27\text{s}$

Die P-Zacke entspricht der Vorhoferregungswelle und kann gewöhnlich bis 0,11 Sekunden breit sein. Die Höhe der P-Zacke schwankt mit dem Alter. Sie sollte jedoch maximal 0,2 mV betragen.

Die Q-Zacke ist sehr schmal und beträgt maximal 0,04 Sekunden. Für die Amplitude gibt es keine absoluten Zahlen. Sie beträgt maximal $\frac{1}{4}$ der jeweiligen R-Zacke.

Die R-Zacke ist unter allen Pulsen die größte Zacke. Sie sollte jedoch 2,6 mV nicht überschreiten.

Die S-Zacke kann unterschiedlich tief werden. Die Dauer beträgt maximal 0,05 Sekunden. Zusammen bilden die Pulse Q,R,S den *QRS-Komplex*. Der Komplex entspricht der Erregungsausbreitung in der Herzkammer und hat eine Dauer von 0,07 - 0,11 Sekunden.

Die T-Zacke entspricht der Erregungsrückbildung in der Kammer und hat eine Amplitude von mindestens $\frac{1}{7}$ der jeweiligen R-Zacke.

4.3. Frequenzspektrum

Unter Frequenzspektrum wird das EKG-Signal verstanden, das durch die EKG-Pulsfolge in einem Abstand von der Herzfrequenz entsteht. Das Frequenzspektrum des EKG-Signals wird benötigt, um in der Signalaufbereitung die notwendigen Filter dimensionieren zu können. Die Modellierung des EKG-Signals mittels $\cos^2(x)$ -Funktion der P-,T-Zacken und mittels Dreiecksfunktion der Q-,R-,S-Zacken erzeugt ein EKG-Signal ohne Störkomponenten, dargestellt in Abbildung 4.3. Das Ziel dabei ist eine Abschätzung des eigentlichen Frequenzspektrums zu erhalten, welches das eigentliche EKG-Signal erzeugt.

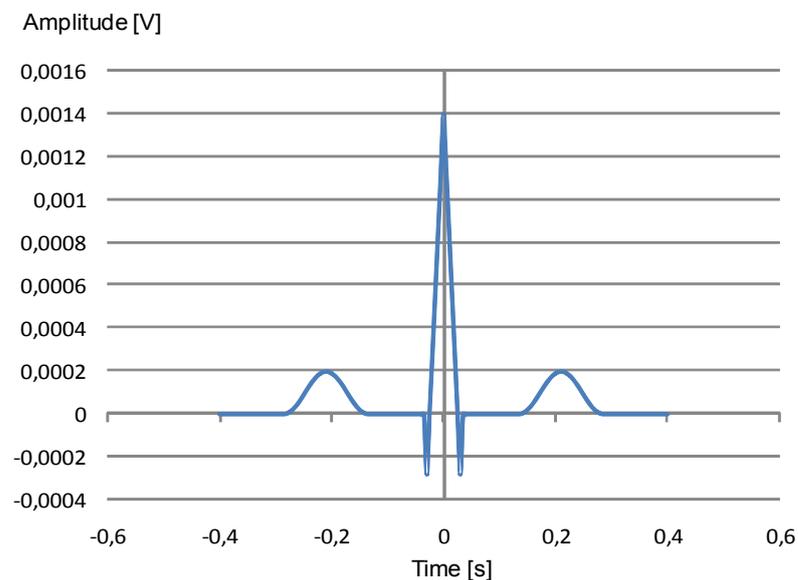


Abbildung 4.3.: EKG-Pulsabschnitt aus Zeitfunktionen zusammengesetzt

Die Zeit- und Amplitudenwerte wurden in Abstimmung mit den Werten in Tabelle 4.1 eingesetzt.

Um auf das Linienspektrum des periodischen EKG-Signals nach Abbildung 4.3 zu kommen, empfiehlt es sich, auf Korrespondenzen²⁴ der Fourier-Transformation

$$s(t) \circ \text{---} \underline{S}(f) \tag{4.1}$$

zurück zu greifen. Anhand des Verschiebungssatzes

²⁴Die Korrespondenzen sind Tabellen in denen zu einer Zeitfunktion die entsprechende Fouriertransformierte Spektralfunktion zu finden ist.

4. Elektrokardiogramm

$$s(t - t_0) \circ\text{---} \underline{S}(f) e^{j2\pi f t_0} \quad (4.2)$$

der Fourier-Transformation können die Funktionen auf der Zeit-Achse an die richtige Position verschoben werden.

Das Spektrum des periodischen EKG-Signals erhält man, indem man die Fourierkoeffizienten $\underline{c}_n(nf_0)$ aus der Spektralfunktion $\underline{S}(f)$ berechnet.

$$\underline{c}_n(nf_0) = \frac{1}{T} \underline{S}(nf_0), \quad f_0 = \frac{1}{T}, \quad n \in \mathbb{Z} \quad (4.3)$$

$\underline{c}_n(nf_0)$ setzt sich aus den Fourier-Koeffizienten a_n und b_n zusammen:

$$\underline{c}_n(nf_0) = \frac{1}{2}(a_n - j b_n) \quad (4.4)$$

Das Signal in Abbildung 4.3 ist symmetrisch zur Y-Achse. Somit fällt der Imaginärteil $b_n = 0$ weg. Daraus ergibt sich das Linienspektrum a_n zu:

$$a_n = 2 \cdot \frac{1}{T} \underline{S}\left(n \frac{1}{T}\right) \quad (4.5)$$

Für die \cos^2 -Funktion $s_{P,T}(t)$ ergeben sich die Spektralfunktionen $\underline{S}_{P,T}(f)$ nach (4.1) und [Herter 2004] zu:

$$\cos^2\left(\frac{\pi t}{2T_i}\right) = \frac{1}{2} \left(1 + \cos\left(\pi \frac{t}{T_i}\right)\right) \circ\text{---} T_i \frac{\text{si}(2\pi f T_i)}{1 - (2f T_i)^2} \quad (4.6)$$

Für die Dreiecksfunktion $s_{Q,R,S}(t)$ ergeben sich die Spektralfunktionen $\underline{S}_{Q,R,S}(f)$ nach (4.1) und [Herter 2004] zu:

$$1 - \frac{|t|}{T_i} \circ\text{---} T_i \text{si}^2(\pi f T_i) \quad (4.7)$$

Das Linienspektrum kann nun folgendermassen mit (4.5) bestimmt werden:

$$a_n = \frac{2}{T} \left(\underline{S}_Q\left(\frac{n}{T}\right) + \underline{S}_R\left(\frac{n}{T}\right) + \underline{S}_S\left(\frac{n}{T}\right) + \underline{S}_{P,T}\left(\frac{n}{T}\right) \underbrace{(e^{-j2\pi f T} + e^{j2\pi f T})}_{2\cos(2\pi f T)} \right) \quad (4.8)$$

4. Elektrokardiogramm

Setzt man für die Zeit- und Amplitudenwerte der einzelnen Zacken realistische Werte für einen Puls von 60 und 120 Schlägen pro Minute ein, so erhält man die Linienspektren in Abbildung 4.4.

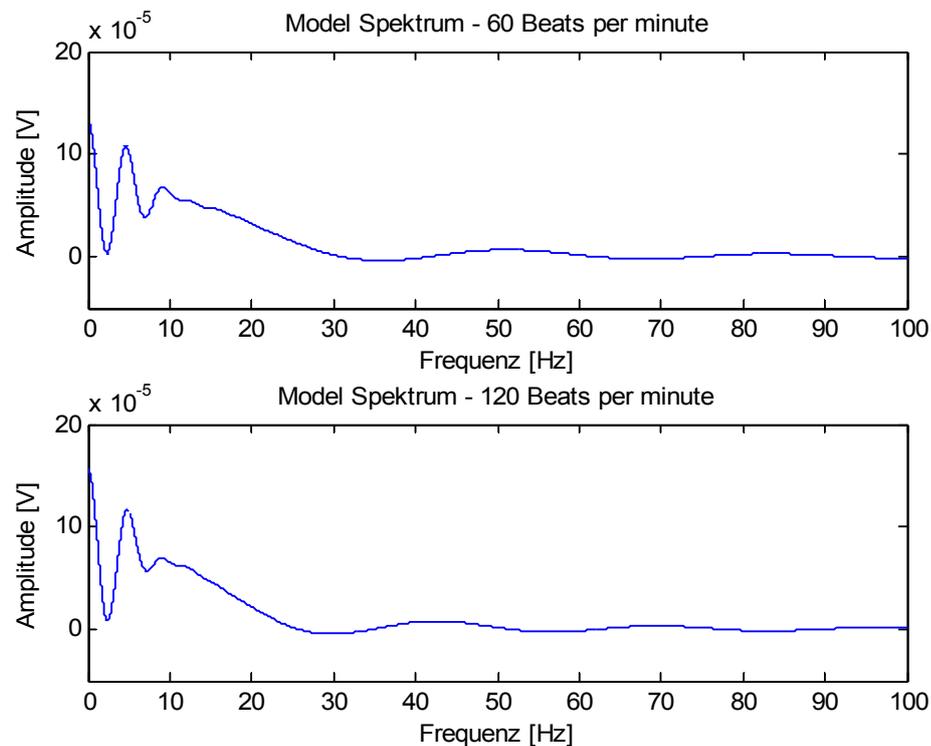


Abbildung 4.4.: Linienspektren der modellierten EKG-Pulsabschnitte

Man erkennt, dass die grössten Amplitudenwerte unterhalb 50Hz aufzufinden sind. Dieses Kenntnis hilft bei dem Vergleich der durch Simulation erzeugten Signale im folgenden Kapitel 4.4. Darüber hinaus dient die Kenntnis der Filterkonzeption in Kapitel 5.

4.4. EKG-Generator

Der EKG-Generator von [EKGGenerator 2009] generiert ein synthetisiertes EKG-Signal mit einstellbarer Herzfrequenz, Anzahl von Herzschlägen, Samplingfrequenz, Wellenform-Morphologie (Timing, Amplitude und Dauer der Zacken), Standardabweichung des RR-Intervalls. Das Rechenmodell basiert auf drei gewöhnlichen Differentialgleichungen, die es ermöglichen, Variationen in der Wellenform-Morphologie und Timing, QT-Zeiten in Abhängigkeit der Pulsfrequenz und der Amplitude von R einzustellen. Die Signalgenerierung

4. Elektrokardiogramm

dient zur Entwicklung von Algorithmen zur biomedizinischen EKG-Signalverarbeitung. Ein generiertes Signal ist in Abbildung 4.5 dargestellt.

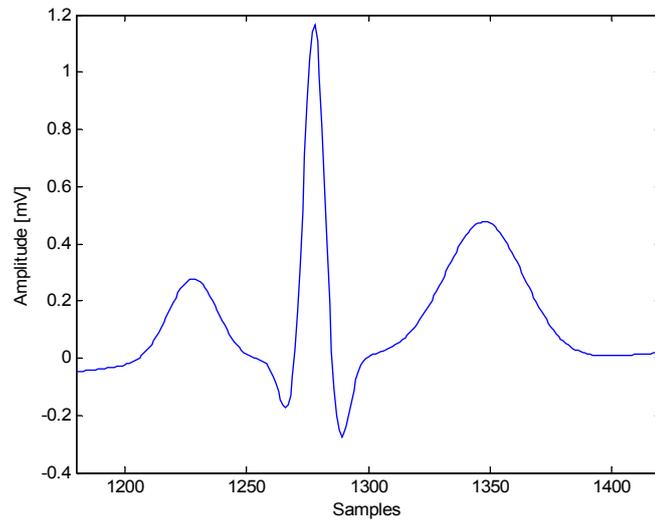


Abbildung 4.5.: EKG-Pulsabschnitt simuliert

Die Berechnung des zugehörigen Spektrums mit Matlab bei einer Herzfrequenz von 120 Schlägen pro Minute zeigt Abbildung 4.6.

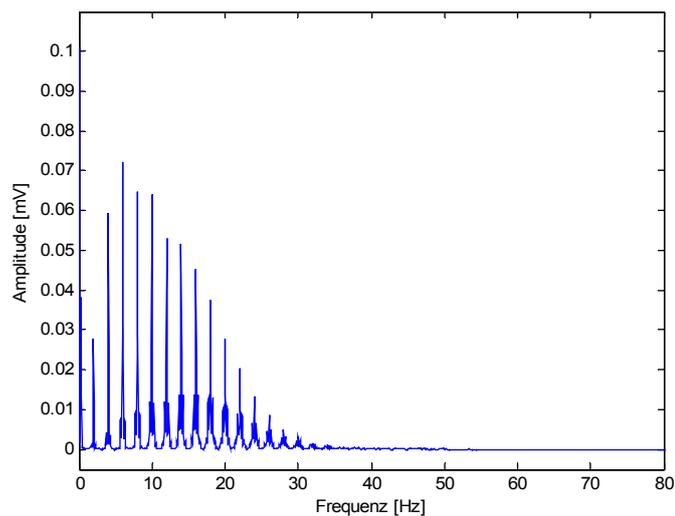


Abbildung 4.6.: Spektrum EKG-Pulsabschnitt simuliert mit einer Herzfrequenz von 120 Schlägen pro Minute

4.5. Resümee

Die Betrachtung der Spektren aus der Fourier-Transformation und der EKG-Generierung zeigen, dass bei einer Herzfrequenz von 120 Schlägen pro Minute die obere Grenzfrequenz beider Spektren zwischen 30Hz und 40Hz liegen. Dies ist eine gute Übereinstimmung zwischen Abschätzung durch die Fourier-Transformation von Zeitfunktionen und Generierung durch Differentialgleichungen in der Simulation.

5. Filterkonzepte

Dieses Kapitel beschäftigt sich mit Filterkonzepten zur Signalaufbereitung. Zur richtigen Konzeption der Filter empfiehlt es sich, zuerst die Anforderungen an die Signalaufbereitung zusammenzufassen. Es folgen die Filterkonzepte und deren Grundlagen zur Unterdrückung bzw. Beseitigung der bereits bekannten Störsignale.

5.1. Anforderungen

Um die Filterung von Störeinflüssen effizient zu gestalten, müssen die entsprechenden Filter für den jeweiligen Störeinfluss dimensioniert werden. Die Art der Störeinflüsse wurde bereits in Kapitel 3 erarbeitet und in Abbildung 3.6 als Domain-Methoden der Störunterdrückung definiert. Die folgende Abbildung 5.1 fasst die Störeinflüsse auf das EKG-Signal in einem Blockdiagramm zusammen.

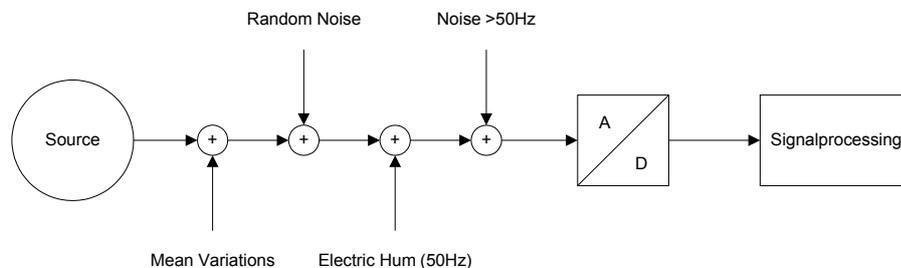


Abbildung 5.1.: Blockschaltbild zur Veranschaulichung der Anforderungen an das Filterkonzept

Ganz links ist die Signalquelle (Source) zu erkennen. Additiv überlagert dazu folgen die Störeinflüsse Mean Variations (Mittelwertschwankungen), Rauschen (Random Noise), Electric Hum (Brummspannung) und Störsignale mit einer Frequenz oberhalb 50Hz. In Kapitel 3.2 wurden die Ursachen zu diesen Störeinflüssen bereits erläutert. Nach der Digitalisierung durch einen A/D-Wandler mit einer Quantisierung von 8 Bit²⁵ sorgt die Signalaufbereitung (Signalprocessing) für eine Unterdrückung der Störeinflüsse.

²⁵Die Quantisierung bezieht sich auf den Amplitudenbereich des EKG-Signals. Der gesamte Messbereich und damit auch die Auflösung des A/D-Wandlers muss grösser gewählt werden, damit die Signalschwankungen bedingt durch die Störeinflüsse erfasst werden können.

5. Filterkonzepte

Fasst man das EKG-Signal mit den Störeinflüssen zusammen, erhält man das Frequenzspektrum in Abbildung 5.2.

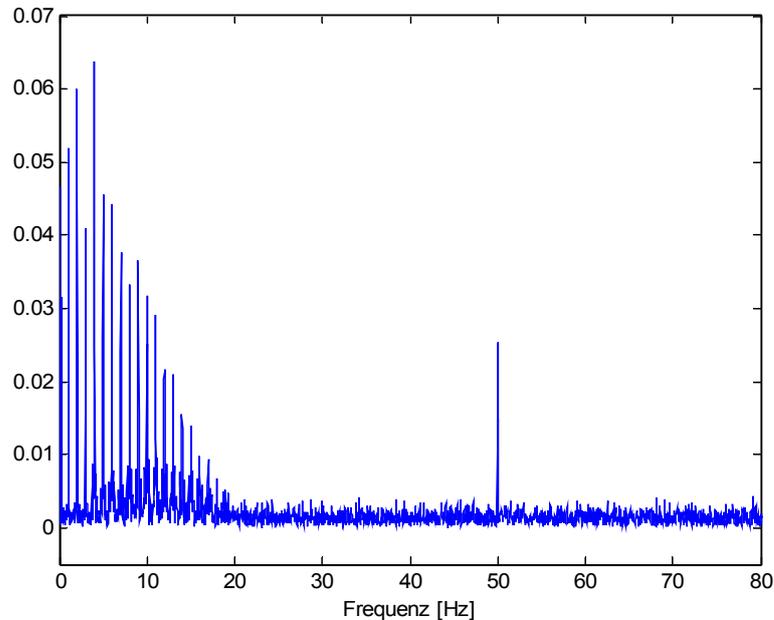


Abbildung 5.2.: Frequenzspektrum des EKG-Signals überlagert mit Störsignalen

Im Bereich 0 bis ca. 30Hz befindet sich der Frequenzbereich des generierten EKG-Signals mit einer Herzfrequenz von 120 Schlägen pro Minute (2Hz). Das Hauptaugenmerk des Frequenzverlaufs in dem Diagramm liegt in dem Rauschteppich, der sich über das ganze Spektrum verteilt und des 50Hz Störers. Es ist klar zu erkennen, dass die Brummfrequenz von 50Hz gut vom EKG-Spektrum getrennt ist. Nachteilig ist, dass das Rauschen auch im EKG-Signalspektrum zu finden ist.

Betrachtet man den Frequenzverlauf in Bezug auf die Samplingrate, so wird deutlich, dass diese im Einklang mit dem Abtasttheorem

$$f_s > 2 \cdot f_{stop} \quad (5.1)$$

grösser $f_s = 100\text{Hz}$ betragen muss. Laut [Krishnan 2007] sollte diese mindestens $f_s = 256\text{Hz}$ betragen.

5.2. Mittelwertschwankungen

Die Ursache der Mittelwertschwankungen liegt in den Bewegungsartefakten der Ableit-elektroden begründet (vgl. Kapitel 3.2). Diese können je nach Art der Ableitung und Person unterschiedlich stark ausfallen. Diese Art der Störung kann auch als Schwankung der Nulllinie²⁶ angesehen werden, die das EKG-Signal überlagert. Abbildung 5.3 zeigt ein EKG-Signal mit dieser Art Störung.

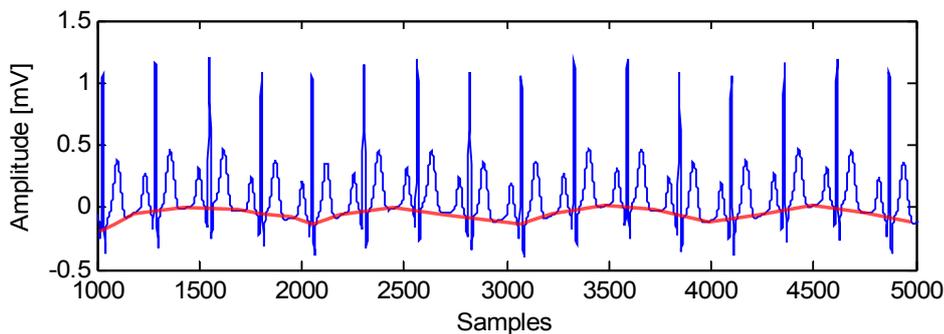


Abbildung 5.3.: Mittelwertschwankungen des EKG-Signals

Die rot eingezeichnete Kurve hebt hierbei den Verlauf der Nulllinie hervor. Die Amplitude der Schwankung kann durchaus auf einen Wert ansteigen, der die Amplitude der R-Zacke übersteigt. Eine Kompensation dieser Störung ist somit notwendig. Ein Mittelwertfilter, der die Kompensation bewerkstelligen kann, zeigt Abbildung 5.4

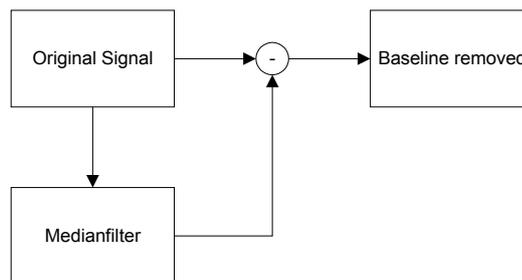


Abbildung 5.4.: Filter zur Kompensation von Schwankungen der Nulllinie (Bildquelle: [Krishnan 2007] modifiziert)

Das Filter berechnet einen adaptiven Offsetwert, der vom Originalsignal abgezogen wird. Das Ergebnis ist ein Signal ohne Schwankungen der Nulllinie. Aus [Krishnan 2007] setzt sich dieses Medianfilter aus einem Block zur Berechnung des Mittelwertes in einem Zeitfenster von 125ms und einem Block, der die Mittelwerte in 500ms Abständen bewertet, zusammen (Abbildung 5.5).

²⁶In der englischen Literatur auch als baseline wander bezeichnet.

5. Filterkonzepte

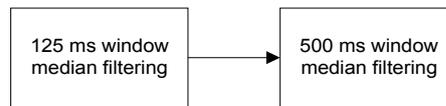


Abbildung 5.5.: Prinzipieller Aufbau des Medianfilters (Bildquelle: [Krishnan 2007] modifiziert)

Die Bestimmung der Mittelwerte A_{mean} ist durch die Gleichung (5.2) beschrieben.

$$A_{mean} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \tag{5.2}$$

Man erhält vier Mittelwerte im Zeitraum von 500ms. Von den vier Werten wird nur derjenige hergenommen, der sich am nächsten zur Nulllinie befindet. Zieht man den so bestimmten Wert vom Signal ab, erhält man ein Signal in Abbildung 5.6, das von Schwankungen der Nulllinie befreit ist.

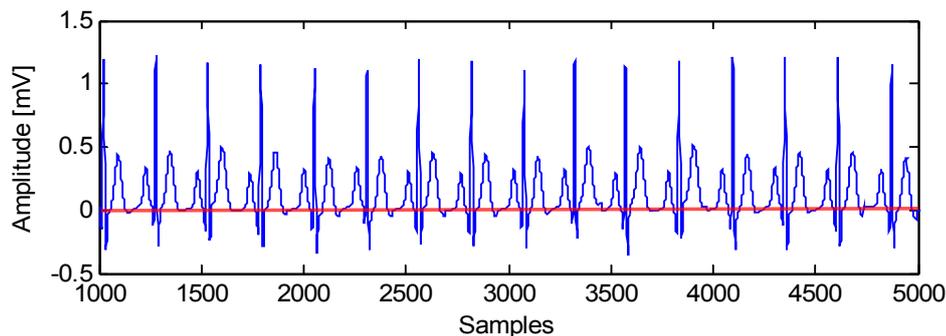


Abbildung 5.6.: Kompensation der Schwankungen der Nulllinie des EKG-Signals

Das Matlab Listing A.2 zu diesem Algorithmus kann im Anhang A.2 eingesehen werden.

5.3. Schätzung von Signalwerten

5.3.1. Autokorrelation

Die Schätzung von Signalwerten, auch als Prädiktion bekannt, ist eine der Standardaufgaben der digitalen Signalverarbeitung [Werner 2008]. Diese Technik bildet die Grundlage zur Bestimmung von Prädiktionsfiltern. Diese Filter bestehen im Prinzip genauso aus der

5. Filterkonzepte

Differenzgleichung nach (3.2). Ausschlaggebend ist die Wahl bzw. Bestimmung der Koeffizienten. Im Fall eines Prädiktionsfilter steht die Berechnung der Erwartungswerte im Mittelpunkt. Der Erwartungswert eines Signals ergibt sich aus:

$$E((x - \mu_x)^2) = \sigma_x^2 = \int_{-\infty}^{\infty} (x - \mu_x)^2 f_x(x) dx \quad (5.3)$$

Wobei σ_x^2 die Varianz darstellt. Ist der Erwartungswert μ_x eines Signals gleich Null²⁷, also $\mu_x = 0$, dann gilt aufgrund der Tatsache $\rho_x^2 = \mu_x^2 + \sigma_x^2$, dass die Varianz σ_x^2 gleich dem quadratischen Mittelwert ρ_x^2 ist. Somit tritt dann folgende Vereinfachung ein:

$$\begin{aligned} E(x^2) &= \sigma_x^2 = \rho_x^2 \quad \text{für zeitkontinuierliche Signale} \\ E(x^2[n]) &= \sigma_x^2[n] = \rho_x^2[n] \quad \text{für zeitdiskrete Signale} \end{aligned} \quad (5.4)$$

Den quadratischen Mittelwert bezeichnet man auch als mittlere Leistung eines Signals. Dies entspricht der Autokorrelationsfunktion $R_{xx}[0]$ an der Stelle 0. Das bedeutet, dass die Erwartungswerte durch die Autokorrelationsfunktion (AKF) bestimmt werden können:

$$E(x[n + l] \cdot x[n]) = R_{xx}[l] \quad (5.5)$$

Die AKF für das EKG-Signal ist in Abbildung 5.7 zu sehen.

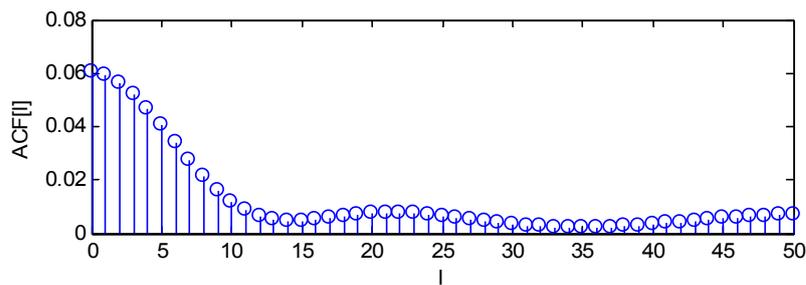


Abbildung 5.7.: Autokorrelationsfunktion des EKG-Signals

²⁷Ist auch als stationäres Signal bekannt. Bedingung für ein stationäres Signal ist die Zeitunabhängigkeit des Mittelwertes.

5.3.2. Wiener-Hopf-Gleichung

Wie bereits erwähnt besteht das Prädiktionsfilter aus der Differenzgleichung nach (3.2). Betrachtet man ein nicht rekursives Filter N-ter Ordnung, so ergibt sich für den Schätzwert der Prädiktion:

$$\hat{x}[n] = \sum_{k=0}^N b_k \cdot x[n-1-k] \quad (5.6)$$

Die Differenz zwischen geschätztem Wert und tatsächlichem Wert wird als Prädiktionsfehler bezeichnet.

$$e[n] = x[n] - \hat{x}[n] \quad (5.7)$$

Der mittlere quadratische Prädiktionsfehler ergibt sich nach (5.4):

$$MSE = E(e^2[n]) = E(x^2[n]) + E(\hat{x}^2[n]) - 2 \cdot E(x[n] \cdot \hat{x}[n]) \quad (5.8)$$

Der Term $E(x^2[n])$ entspricht dabei der AKF an der nullten Stelle $R_{xx}[0]$. Zur Berechnung der Filterkoeffizienten bietet sich das Kriterium des kleinsten mittleren quadratischen Fehlers MSE an. Dazu setzt man für $\hat{x}[n]$ die Gleichung (5.6) ein und bestimmt die partielle Ableitung des MSE.

$$\frac{d}{db_k} MSE = 2 \cdot \sum_{l=0}^N b_l \cdot R_{xx}[l-k] - 2 \cdot R_{xx}[k+1] \stackrel{!}{=} 0 \quad (5.9)$$

Man erhält schliesslich nach [Werner 2008] die Matrix für ein Prädiktionsfilter mit beispielweise 3.Ordnung:

$$\begin{pmatrix} R_{xx}[0] & R_{xx}[1] & R_{xx}[2] \\ R_{xx}[1] & R_{xx}[0] & R_{xx}[1] \\ R_{xx}[2] & R_{xx}[1] & R_{xx}[0] \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} R_{xx}[1] \\ R_{xx}[2] \\ R_{xx}[3] \end{pmatrix} \quad (5.10)$$

Diese Korrelationsmatrix ist eine symmetrische Toeplitz-Matrix. Stellt man die Matrixgleichung nach b um erhält man folgende Gleichung, die auch als Wiener-Hopf-Gleichung bekannt ist:

$$b_{opt} = R_{xx}^{-1} \cdot r_{xx} \quad (5.11)$$

5.3.3. Levinson-Durbin-Algorithmus

Im Levinson-Durbin-Algorithmus werden die Prädiktionskoeffizienten in rekursiver Weise mit zunehmender Ordnung aus (5.10) berechnet. Matlab bietet die Funktion *levinson* an, die Koeffizienten für Lattice-²⁸ und IIR-Filter Strukturen liefert. Die Herleitung des Algorithmus wird der Literatur [Werner 2008] überlassen. Der folgende mathematische Zusammenhang zeigt die Berechnung der Hilfsgrößen und der Prädiktionskoeffizienten. Aus (5.10) sind folgende Vereinfachungen für AKF und die MSE-optimale Koeffizienten des Prädiktors p-ter Ordnung notwendig:

$$\begin{aligned} r_i &= R_{xx}[i] \\ \alpha_i^{(p)} &= -b_{i-1}^{(p)} \quad \text{für } i = 1, \dots, p \end{aligned} \quad (5.12)$$

Damit ergeben sich für den Algorithmus folgende Berechnungen:

$$\begin{aligned} q^{p-1} &= r_p + \sum_{m=1}^{p-1} \alpha_m^{(p-1)} \cdot r_{p-m} \\ E^{(p-1)} &= r_0 + \sum_{m=1}^{p-1} \alpha_m^{(p-1)} \cdot r_m \\ k_p &= -\frac{q^{(p-1)}}{E^{(p-1)}} \\ E^{(p)} &= E^{(p-1)} \cdot (1 - k_p^2) \end{aligned} \quad (5.13)$$

Die Prädiktionskoeffizienten zur p-ter Ordnung sind im folgenden aus (5.12) bestimmt:

$$\begin{aligned} \alpha_0^{(p)} &= 1 \\ \alpha_m^{(p)} &= \alpha_m^{(p-1)} + k_p \cdot \alpha_{p-m}^{(p-1)} \quad \text{für } m = 1 : p-1 \text{ und } p \geq 2 \\ \alpha_p^{(p)} &= k_p \end{aligned}$$

Die Übertragungsfunktion $H_e(z)$ des Prädiktionsfilters verdeutlicht an dieser Stelle nochmals den Bezug zum nicht rekursiven Filter nach Gleichung (5.6) und (5.7).

²⁸Lattice-Strukturen stellen alternative Formen zur Implementierung digitaler LTI-Systeme bereit. Ihr modularer Aufbau lässt eine adaptive und stufenweise Erweiterung zu.

5. Filterkonzepte

$$H_e(z) = 1 - z^{-1} \cdot \sum_{m=0}^{p-1} b_m z^{-m} = \sum_{m=0}^p \alpha_m z^{-m} \quad (5.14)$$

Das Listing A.3 im Anhang A.3 zeigt eine C-Implementierung des Algorithmus. Zusammenfassend lässt sich die Generierung der Koeffizienten zur Bestimmung eines Prädiktionsfilters in folgendem Blockdiagramm in Abbildung 5.8 darstellen.

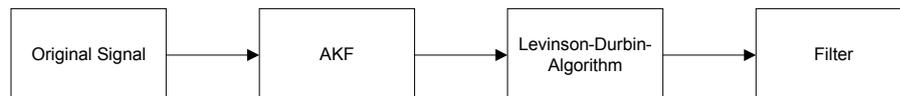


Abbildung 5.8.: Generierung von Filterkoeffizienten zur Schätzung von Signalwerten

5.3.4. Autoregressive Spektralschätzung

Die Schätzung von Signalwerten bildet die Grundlage der Autoregressiven Modellierung bzw. Autoregressive Spektralschätzung. Anwendung findet die Modellierung in der Medizintechnik häufig zur Bestimmung der Herzfrequenzvariabilität (HRV). Die Autoregressive Spektralschätzung (AR) zählt zu den modernen Schätzmethoden der spektralen Leistungsdichte. Hierbei handelt es sich bei der autoregressiven Schätzung um ein modellgestütztes Spektralschätzverfahren. Als Modell dient dabei ein lineares System, das mit weißem Rauschen erregt wird und dessen Ausgangssignal möglichst gleiche Spektraleigenschaften aufweist, wie die des zu analysierenden Signals [Krishnan 2007]. Das zugrunde liegende Signal wird also durch die Systemparameter beschrieben, weshalb dieses Verfahren unter anderem auch als parametrisches Schätzverfahren bezeichnet wird. Abbildung 5.9 zeigt den prinzipiellen Aufbau des Filters.

5. Filterkonzepte

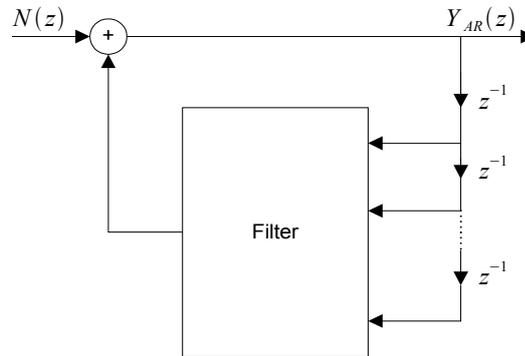


Abbildung 5.9.: Blockschaltbild des Filters zur AR Modellierung

Hierbei handelt es sich um ein IIR-Filter zur AR-Modellierung mit der Übertragungsfunktion

$$H_{AR}(z) = \frac{Y_{AR}(z)}{N(z)} = \frac{1}{1 + \sum_{i=0}^N a_i z^{-i}} \quad (5.15)$$

Durch eine genügend große Filterordnung²⁹ des Prädiktionsfilters nach Gleichung (5.6) entspricht der Prädiktionsfehler etwa dem Weissen Rauschen mit der Leistung σ_e^2 . Nimmt man nun als Eingangssignal $N(z)$ des IIR-Filters (AR-Modell), weißes Rauschen mit der $\sigma_e^2 = \sigma_N^2$, so hat dessen Ausgangssignal $Y_{AR}(z)$ die gleichen statistischen Eigenschaften wie das Originalsignal.

Mit der Übertragungsfunktion $H_{AR}(z)$ und dem Weissen Rauschen σ_N^2 erhält man nach [Grueningen 2002] das geschätzte Ausgangsleistungsdichtespektrum

$$\hat{S}_{AR}(e^{j\Omega}) = |H_{AR}(e^{j\Omega})|^2 \sigma_N^2 \quad (5.16)$$

Es gilt zu berücksichtigen, dass das IIR-Filter instabil werden kann. Die Verwendung des Burg-Algorithmus³⁰ hingegen liefert gegenüber dem Levinson-Durbin-Algorithmus ein stabiles AR-Modell [Krishnan 2007].

²⁹Nach [Krishnan 2007] sind hierfür 32 Koeffizienten nötig.

³⁰Der Burg-Algorithmus berücksichtigt sowohl die Vorwärts- als auch die Rückwärts-Prädiktionsfehlerleistung zur Schätzung des Leistungsdichtespektrums. Die Schätzung der Filterparameter erfolgt hierbei nicht über die Berechnung der Korrelationsmatrix sondern direkt von den Daten.

5.4. Rauschunterdrückung

Eine oft verwendete Möglichkeit der Rauschunterdrückung basiert auf dem LMS-Algorithmus (Least Mean Square). Hierbei wird versucht, den mittleren quadratischen Fehler MSE in Gleichung (5.8) zwischen einem Referenzsignal und dem Eingangssignal zu minimieren. Jedoch besteht oft das Problem, ein geeignetes Referenzsignal zu generieren [Krishnan 2007].

Morphologische Operatoren hingegen sind in der Signal- und Bildverarbeitung, wegen ihrer Robustheit und ihren adaptiven Fähigkeiten weit verbreitet. Es gibt 2 grundlegende morphologische Operatoren: Erosion (\ominus) und Dilation (\oplus). Darauf aufbauend gibt es 2 weitere Operatoren: Opening (\circ) und Closing (\bullet), diese sind aus Erosion und Dilation abgeleitete Operatoren. Im folgenden werden die Operatoren im Detail vorgestellt, wobei $X[n]$, $n=0,1,\dots,N-1$ ein diskretes Signal und $B[m]$, $m=0,1,\dots,M-1$ ein symmetrisches Strukturelement³¹ darstellen.

$$(X \ominus B)[n] = \min_{m=0,\dots,M-1} \left\{ X\left[n - \frac{M-1}{2} + m\right] - B[m] \right\} \quad (5.17)$$

Erosion ist ein „shrinking“ Operator bei dem die Werte $\underline{X \ominus B}$ immer kleiner als die Werte von \underline{X} sind.

$$(X \oplus B)[n] = \max_{m=0,\dots,M-1} \left\{ X\left[n - \frac{M-1}{2} + m\right] - B[m] \right\} \quad (5.18)$$

Dilation ist ein „expansion“ Operator bei dem die Werte von $\underline{X \oplus B}$ immer grösser als die Werte von \underline{X} sind. Für Opening und Closing gilt:

$$\underline{X} \circ \underline{B} = (\underline{X} \ominus \underline{B}) \oplus \underline{B} \quad (5.19)$$

$$\underline{X} \bullet \underline{B} = (\underline{X} \oplus \underline{B}) \ominus \underline{B} \quad (5.20)$$

Beim Opening bleiben die Strukturen des Signals erhalten, die durch das Strukturelement \underline{B} beschrieben werden. Wie der Name Opening schon sagt, werden zusammengesetzte Strukturen an ihren „Schwachstellen“ geöffnet. Übertragen auf eine Datensequenz neigt das Opening zu den höheren Werten, die durch das Strukturelement \underline{B} bestimmt werden. Beim Closing lassen sich lokal begrenzte Störungen des Signals unterdrücken bzw. kleine Strukturen gezielt herausfiltern. Closing neigt dazu, wie der Name schon erkennen lässt,

³¹Wird häufig auch als morphological Mask bezeichnet, da die Positionen in der Maske unterschiedlich stark gewichtet werden können.

5. Filterkonzepte

Strukturen zu schliessen³². Damit man mit den morphologischen Operatoren eine Rauschunterdrückung erreicht, wird aus ihnen eine Art Mittelwert \bar{X} gebildet, das die folgende Gleichung veranschaulicht:

$$\bar{X} = \frac{1}{2} [\underline{X} \bullet \underline{B} + \underline{X} \circ \underline{B}] = \frac{1}{2} [(\underline{X} \oplus \underline{B}_1) \ominus \underline{B}_2 + (\underline{X} \ominus \underline{B}_1) \oplus \underline{B}_2] \quad (5.21)$$

Wie man aus dieser Gleichung erkennen kann, verwendet man verschiedene Strukturelemente \underline{B} . Für $\underline{B}_1 = (0, 0.004, 0.02, 0.004, 0)$ hat sich ein Vektor mit Dreiecksfunktion und für $\underline{B}_2 = (0, 0, 0, 0, 0)$ ein Vektor mit einer Geraden bewährt. Die Länge des Strukturelements \underline{B} hängt von der Abtastrate ab und wird auf eine Anzahl von 5 Elementen festgelegt [Krishnan 2007]. Übertragen auf das EKG-Signal zeigt Abbildung 5.10 die Anwendung der morphologischen Filterung.

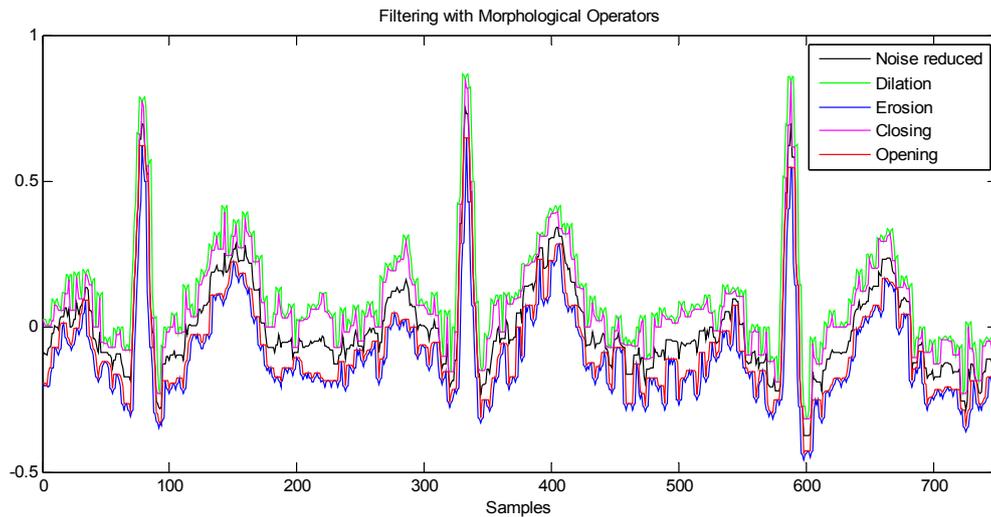


Abbildung 5.10.: Arbeitsweise der Filterung mit Morphologischen Operatoren

Die schwarze Kurve zeigt den Mittelwert \bar{X} .

Zur Verdeutlichung zeigt Abbildung 5.11 oben das Signal vor der Filterung und die Abbildung 5.11 unten das Signal nach der Filterung. Das Listing A.4 im Anhang A.4 zeigt die Umsetzung der Rauschunterdrückung in Matlab.

³²Man kann sich Opening und Closing auch bildlich vorstellen, indem man sich zwei schwarz nebeneinander liegende ausgefüllte Kreise vorstellt die sich nicht berühren. Die Farbe schwarz hat hierbei einen kleinen und die Farbe weiss einen hohen Wert. Mit Closing werden die Kreise vergrößert, so dass sie sich verbinden können (das Ergebnis ist ein kleinerer Bildmittelwert). Mit Opening werden die Kreise im Durchmesser kleiner, so dass der Abstand sich zwischen ihnen vergrößert (das Ergebnis ist ein grösserer Bildmittelwert).

5. Filterkonzepte

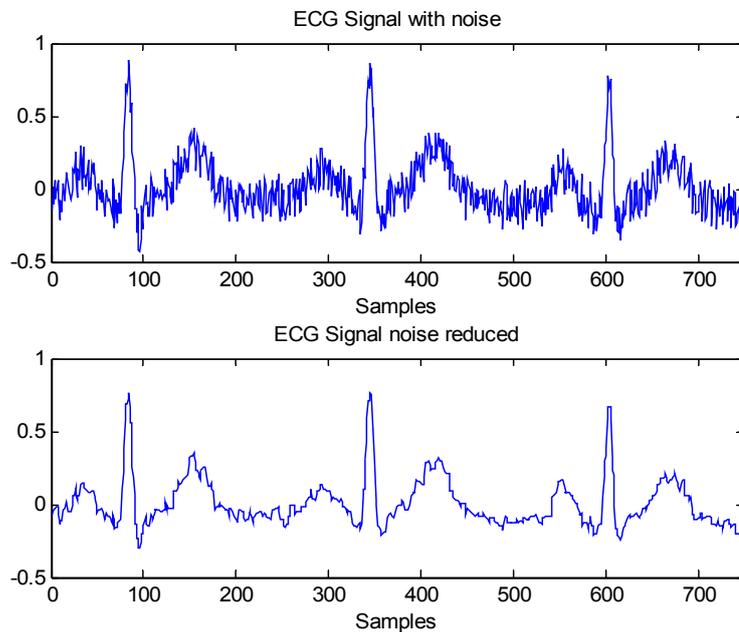


Abbildung 5.11.: Rauschunterdrückung des gestörten EKG-Signals mittels morphologischen Operatoren. Oben das Signal vor der Filterung. Unten das Signal nach der Filterung

5.5. Dämpfung höherfrequenter Signalanteile

5.5.1. Filterspezifikation

Zur Unterdrückung der Brummfrequenz von 50Hz und darüber liegende Störsignale wird ein „statisches“ Filter eingesetzt. Ziel ist es, die ungewünschten Störanteile zu beseitigen, ohne das EKG-Signal zu dämpfen. Aus den bisherigen Betrachtungen ist hierfür ein Tiefpassfilter am besten geeignet. Die Filterspezifikation zeigt Abbildung 5.12.

Der Durchlassbereich befindet sich von 0 - 31Hz. Es soll ein IIR-Filter mit Kaskadenstruktur³³ zum Einsatz kommen. Diese Struktur hat den Vorteil, dass sie mit einer minimalen Anzahl von Verzögerungselementen implementiert werden kann.

³³Wegen ihrer guten Eigenschaft bezüglich der Stabilität ist die Kaskade 2.Ordnung die häufigste Struktur zur Realisierung von IIR-Filtern.

5. Filterkonzepte

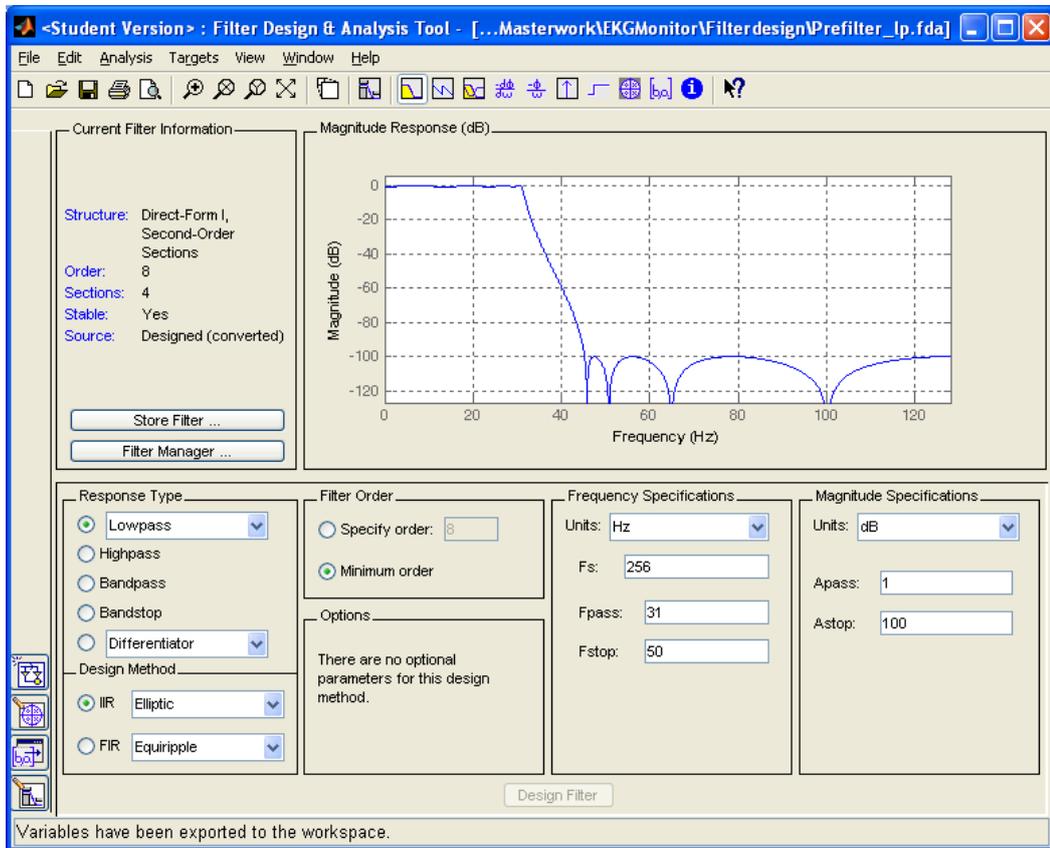


Abbildung 5.12.: Matlabs FDATool zur Filterspezifikation

5.5.2. Filterentwurf

Aus dem Matlab Tool *fdatool* ergibt sich ein Filter 8. Ordnung mit 4 Blöcken der Direktform-II-Struktur, wie in Abbildung 5.13 dargestellt.

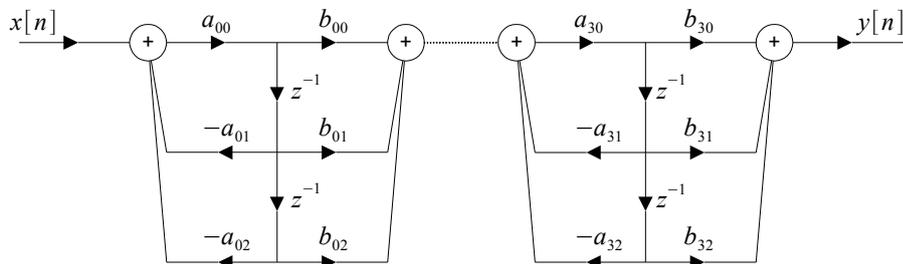


Abbildung 5.13.: Filter 8. Ordnung mit Blöcken zweiter Ordnung in Direktform-II-Struktur (Bildquelle: [Grueningen 2002] modifiziert)

5. Filterkonzepte

Die Quantisierung der Filterkoeffizienten beträgt 8Bit. Das entsprechende Matlab-Skript zur Skalierung und Koeffizientenquantisierung befindet sich im Listing A.5 im Anhang A.5. Zur Überprüfung des Filterentwurfes kann das Matlab Tool *fvtool* aufgerufen werden. Das Ergebnis ist die Übertragungsfunktion in Abbildung 5.14 und das Pol-, Nullstellendiagramm in Abbildung 5.15.

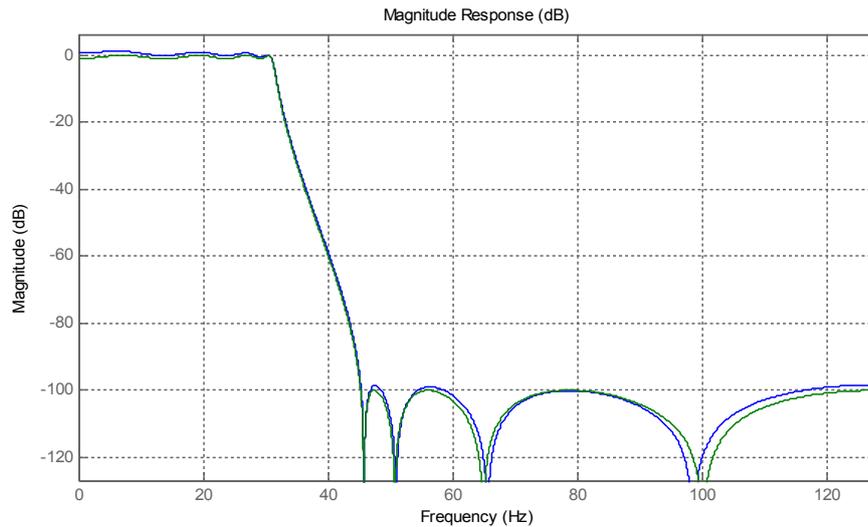


Abbildung 5.14.: Übertragungsfunktion des Tiefpassfilters

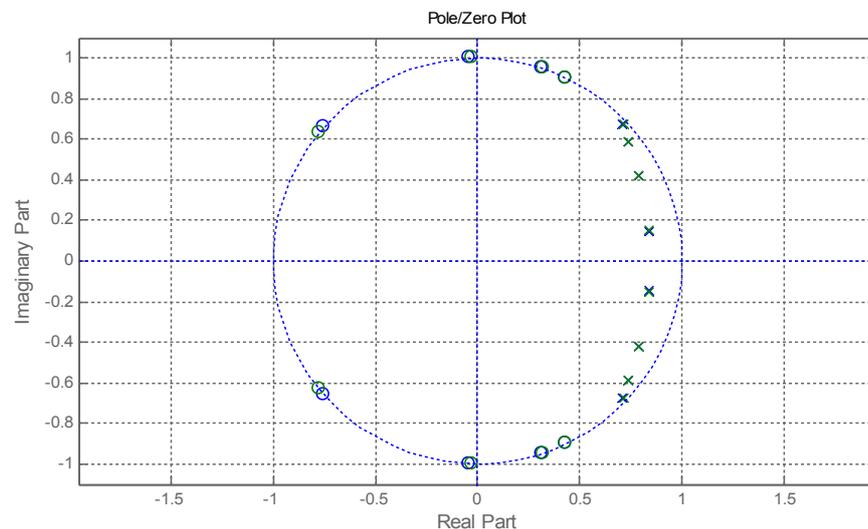


Abbildung 5.15.: Pol-, Nullstellendiagramm des Tiefpassfilters

Die grüne Kurve kennzeichnet das Filterverhalten mit den quantisierten Koeffizienten. In dem Pol-, Nullstellendiagramm befinden sich alle Pole innerhalb des Kreises, das Filter ist somit stabil. Die quantisierten Koeffizienten sind in Tabelle 5.1 aufgelistet.

5. Filterkonzepte

Tabelle 5.1.: Quantisierte Koeffizienten für das Filter mit Blöcken 2. Ordnung in Direktform-II-Struktur

Block	b_{0x}	b_{1x}	b_{2x}	a_{0x}	a_{1x}	a_{2x}
x=1	0.0156	0.0234	0.0156	1.0000	-1.6797	0.7266
x=2	0.1016	0.0078	0.1016	1.0000	-1.5781	0.7969
x=3	0.2969	-0.1875	0.2969	1.0000	-1.4688	0.8828
x=4	0.4609	-0.3984	0.4609	1.0000	-1.4219	0.9609

Die erste Zeile zeigt die Koeffizienten des ersten Filterblockes, die zweite Zeile den zweiten Block, etc.

5.6. Blockschaltbild

Abschließend können die in diesem Kapitel besprochenen Filterkonzepte in einem Blockschaltbild in Abbildung 5.16 dargestellt werden.



Abbildung 5.16.: Blockschaltbild der Signalaufbereitung

Der erste Block unterdrückt das Rauschen. Der nachfolgende Block beseitigt die Schwankungen der Nulllinie. Der letzte Block schliesslich vermindert bzw. beseitigt die Störanteile mit Frequenzen $\geq 50Hz$. Nachdem das EKG-Signal mit seinen Störanteilen die Filterstrecke durchlaufen hat, kann folgendes Frequenzspektrum in Abbildung 5.17 beobachtet werden.

5. Filterkonzepte

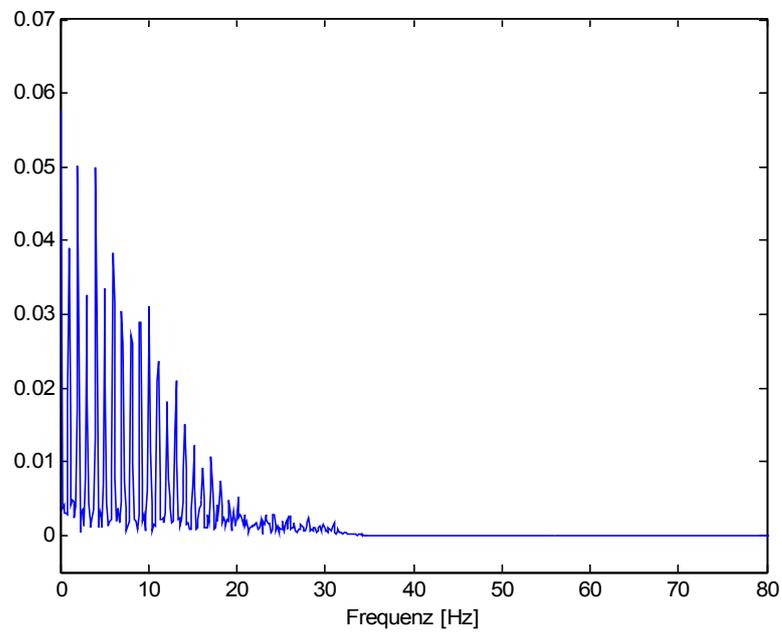


Abbildung 5.17.: Frequenzspektrum nach der Signalaufbereitung

Gegenüber Abbildung 5.2 zeigt sich, dass die höherfrequenten Anteile verschwunden sind. Ebenso der Rauschteppich.

6. EKG-Signalkompression

In diesem Kapitel werden die Techniken zur EKG-Signalkompression vorgestellt. Eine Technik, die dafür eingesetzt wird, ist die Prädiktion. Diese baut auf dem Thema Schätzung von Signalwerten, wie in Kapitel 5.3 beschrieben, auf. Die Prädiktion dient hierbei zur Dekorrelation und zur Vorbereitung der Entropiecodierung. Das dafür eingesetzte Verfahren der Huffman-Codierung und die Generierung der benötigten Huffman-Codes werden beschrieben. Abschließend folgt zur Übersicht das Blockschaltbild der EKG-Signalkompression.

6.1. Dekorrelation

6.1.1. Lineare Prädiktion

Eine Methode um Dekorrelation zu erreichen, also Information zu konzentrieren, ist die Prädiktion. Grundlage hierfür ist die Schätzung von Signalwerten (vgl. Kapitel 5.3). Bei der Schätzung entsteht ein Prädiktionsfehler $e[n]$. Dieser wird bei groß gewählter Filterordnung näherungsweise zu Weisssem Rauschen. Weisses Rauschen ist unkorreliert, die zugehörige AKF dazu lautet:

$$R_{ee}[l] = \begin{cases} \sigma_e^2 & \text{für } l=0 \\ 0 & \text{sonst.} \end{cases}$$

Die Rauschleistung hierbei beträgt σ_e^2 . Die Generierung des Prädiktionsfehlers als dekorreliertes Signal von $x[n]$ zeigt folgende Abbildung 6.1.

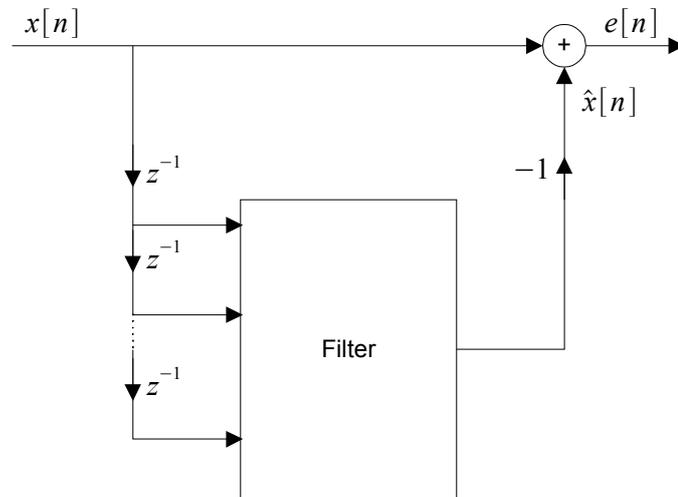


Abbildung 6.1.: Blockschaltbild für die lineare Prädiktion

Es handelt sich hierbei um eine FIR-Filter. Die Filterkoeffizienten werden, wie in Kapitel 5.3 beschrieben, durch den Levinson-Durbin-Algorithmus bestimmt.

Als wichtige Kenngröße der Leistungsfähigkeit der Prädiktion wird das Verhältnis der Leistungen des Signals $x[n]$ und des Fehlersignals $e[n]$ verwendet. Es entsteht der Prädiktionsgewinn³⁴ G_p .

$$G_p = \frac{R_{xx}[0]}{R_{ee}[0]} \quad (6.1)$$

³⁴Beträgt der Gewinn gleich 4 (6dB), so entspricht dies, im Vergleich zum ursprünglichen Signal, einer mittleren Abnahme der Signalamplitude des Fehlersignals um den Faktor 2. Das ermöglicht wiederum eine Reduktion der zur Signalwertdarstellung verwendeten Wortlänge um 1 Bit bei etwa gleicher Qualität [Werner 2008].

6.1.2. Prädiktion mit Quantisierung

Für die weitere Verarbeitung bzw. zur Datenübertragung des Prädiktionsfehlers muss dieser ganzzahlig sein. Dies erreicht man mit einer Quantisierung. Man erhält einen quantisierten Prädiktionsfehler $e_Q[n]$. Die Abbildung 6.2 zeigt das zugehörige Blockschaltbild.

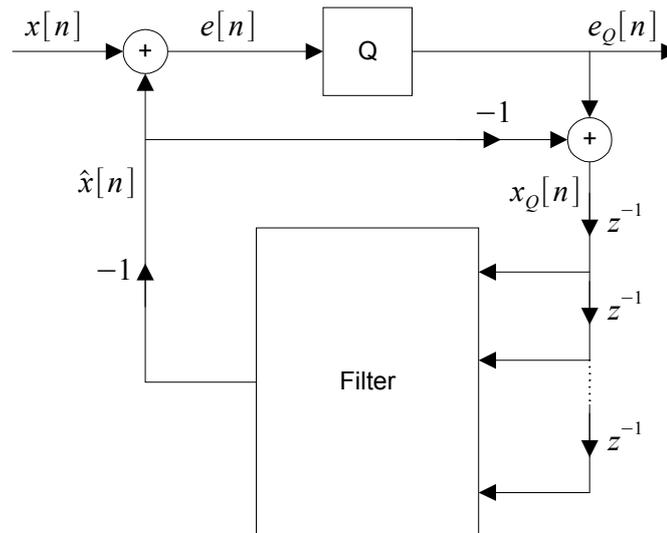


Abbildung 6.2.: Blockschaltbild für die lineare Prädiktion mit Quantisierung

Auf den ersten Blick scheint es sich hierbei um einen IIR-Filter zu handeln. Bei genauerem hinsehen erkennt man, dass der Schätzwert $\hat{x}[n]$ auf den Eingang zurückgeführt wird. Dadurch erhält man wieder die gewohnte FIR-Struktur wie in Abbildung 6.1. Angewendet auf das EKG-Signal erhält man in Abbildung 6.3 den Verlauf des Prädiktionsfehlers für Prädiktion 2. Ordnung und in Abbildung 6.4 den Verlauf für Prädiktion 5. Ordnung.

6. EKG-Signalkompression

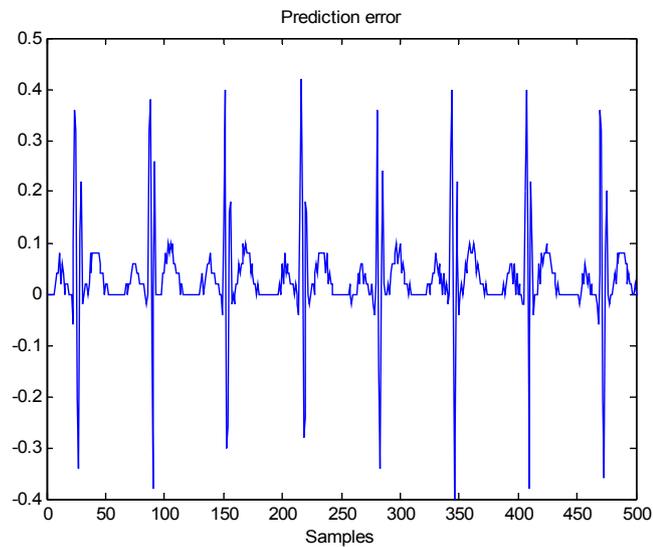


Abbildung 6.3.: Prädiktionsfehler mit Prädiktion 2.Ordnung

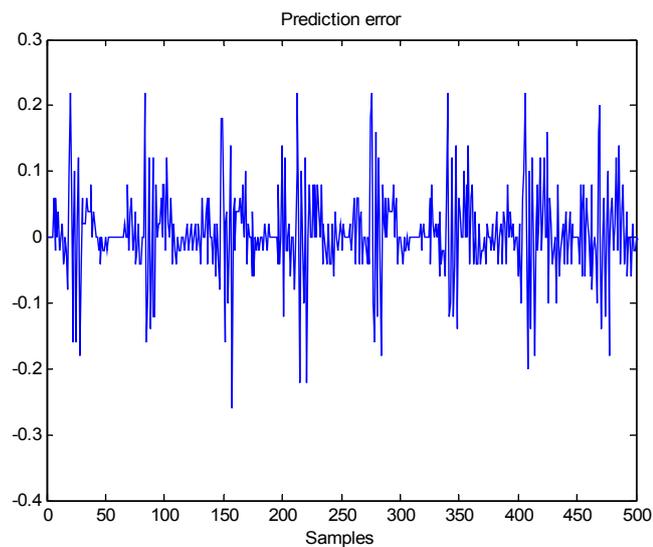


Abbildung 6.4.: Prädiktionsfehler mit Prädiktion 5.Ordnung

Man erkennt, dass, je höher die Ordnung gewählt wird, das Signal immer näher dem Weissen Rauschen ähnelt und somit die Korrelation immer weiter vermindert wird. Darüber hinaus sind deutlich Signalspitzen zu erkennen, die durch die R-Zacke verursacht werden. Zur weiteren Dynamikreduktion des Prädiktionsfehlersignals kann dies ausgenutzt wer-

6. EKG-Signalkompression

den. Dazu wird die aktuelle Signalperiode aus einer vorhergehenden Periode geschätzt³⁵. Für die Filterordnungen 2 bis 5 resultiert der Prädiktionsgewinn mit der Gleichung (6.1) zu folgenden Werten, die in Tabelle 6.1 dargestellt sind.

Tabelle 6.1.: Prädiktionsgewinn in Abhängigkeit der Prädiktorordnung

Ordnung	Gewinn
2	7,2dB
3	8,9dB
4	10,1dB
5	10,2dB

Mit diesen Werten ist eine Reduktion der Signalwertdarstellung um 2Bit bei Filterordnung 4 und 5 mit „wenig“ Verlust und bei Filterordnung 2 mit „mehr“ Verlust³⁶ möglich. Eine weitere interessante „Kenngröße“ des Prädiktors ist der mittlere quadratische Fehler (MSE) der Prädiktion. Mit der Korrelationsmatrix R_{xx} in (5.10) ergibt sich laut [Werner 2008] der MSE zu:

$$MSE = R_{xx}[0] + b^T \cdot R_{xx} \cdot b - 2 \cdot b^T \cdot r_{xx} \quad (6.2)$$

Setzt man die Wiener-Hopf-Gleichung (5.11) ein, so erhält man den MMSE-optimalen Prädiktor:

$$MSE_{opt} = R_{xx}[0] - b_{opt}^T \cdot r_{xx} \cdot b \quad (6.3)$$

Schliesslich erhält man für den Prädiktor 2.Ordnung folgende Gleichung:

$$MSE_{opt} = R_{xx}[0] - b_0 \cdot R_{xx}[1] - b_1 \cdot R_{xx}[2] \quad (6.4)$$

Wendet man den Prädiktor 2.Ordnung für das EKG-Signal an, so ergeben sich die Werte für den mittleren quadratischen Fehler $MSE_{opt} = 0,0078$ und für die Koeffizienten: $b_0 = -1,3501$ und $b_1 = 0,6494$. Die Abhängigkeit des MSE_{opt} von den Koeffizienten b_0 und b_1 zeigt die Abbildung 6.5.

³⁵Bei Kenntnis der momentanen Periodenlänge wird die Differenz im Sinne eines zweiten Prädiktionsfehlersignals berechnet. Diese zweite Prädiktion wird wegen der „längerfristigen“ Vorhersage auch als Langzeitprädiktion bezeichnet.

³⁶Bei 9dB ist die Reduktion von einem Bit ohne Verlust möglich.

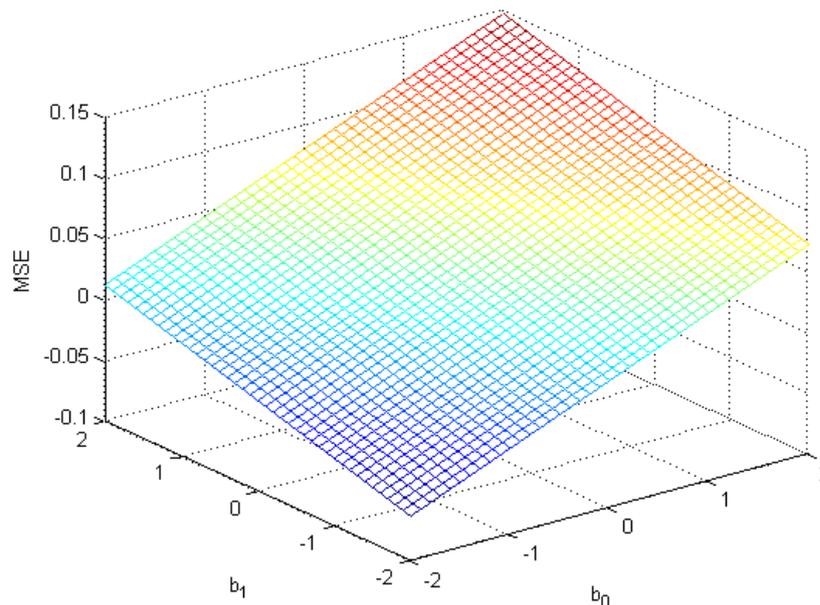


Abbildung 6.5.: Abhängigkeit des MMSE-optimalen Prädiktors 2.Ordnung von seinen Koeffizienten

6.2. Entropiecodierung

6.2.1. Huffman-Codierung

Die Huffman-Codierung gehört zur Gruppe der verlustlosen Kompression. Hierbei unterscheiden sich die decodierten Daten nicht von den Originaldaten. Grundlage der Entropiecodierung ist die Auftretenswahrscheinlichkeit p_i eines Zeichens, man erhält damit den Informationsgehalt des Zeichens x_i :

$$I(x_i) = \log_2\left(\frac{1}{p_i}\right) = -\log_2(p_i)$$

Für eine Datenquelle als Ganzes definiert man dann die sogenannte Entropie H (3.5) als den gewichteten Durchschnitt der Informationsgehalte aller Symbole x_i . Diese Entropieformel gibt die theoretisch optimal mögliche Codierung an, also in Bezug auf die Datenkompression die kleinst mögliche Codewortlänge. Betrachtet man in diesem Zusammenhang die Auftretenswahrscheinlichkeit des quantisierten Prädiktionsfehlers 4. Ordnung, so kann man folgende Verteilung in Abbildung 6.6 beobachten:

6. EKG-Signalkompression

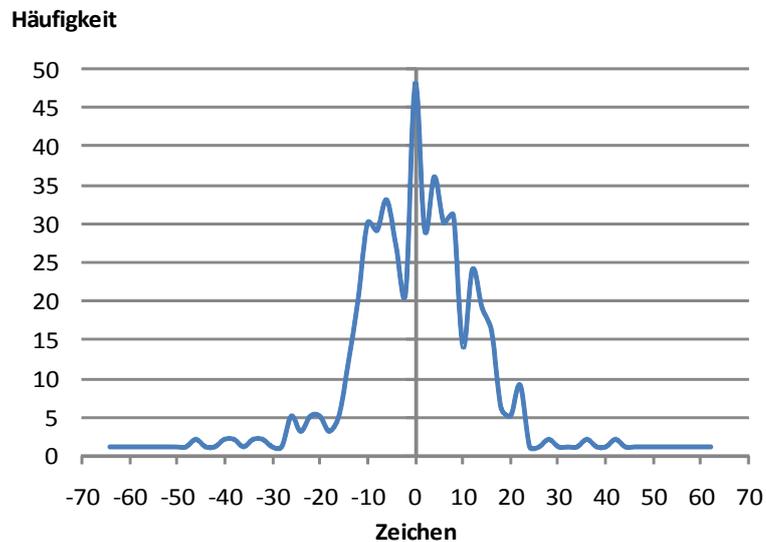


Abbildung 6.6.: Häufigkeitsverteilung der zu codierenden Zeichen

Die x-Achse zeigt hierbei die Amplitudenwerte des Prädiktionsfehlersignals, multipliziert mit dem Faktor 100, um ganzzahlige Werte zu erhalten. Für diese Verteilung ergibt sich die Entropie aus Gleichung (3.5) zu $H(z)=4,83\text{Bit}$ pro Zeichen. Dies bedeutet, man benötigt im Mittel 4,83Bit, um die Zeichen zu codieren. Dies steht der maximalen Entropie von 6Bit pro Zeichen bei einer Gleichverteilung gegenüber. Stellt man die Effizienz η

$$\eta = \frac{H(x)}{L} = \frac{H(x)}{\sum_{i=1}^q p(x_i) \cdot m_i} \tag{6.5}$$

für die Datenübertragung ohne Entropiecodierung auf, entsteht ein Wert von $\eta = 0,805$ (80%) bei einer Codewortlänge $m_i = 6 \mid \forall i \in \mathbb{N}$. Mit Entropiecodierung ($H(x)=4,83\text{Bit}/\text{Zeichen}$) steigt die Effizienz auf $\eta = 0,995$ (99%).

6.2.2. Codegenerierung

Mit der Huffman-Codierung kann ein optimaler, eindeutig decodierbarer Code erzeugt werden. Dies ist grundlegend, um bei der Decodierung wieder auf die ursprünglichen Daten zurückzukommen³⁷. Die Erzeugung von Huffman-Code zeigt Abbildung 6.7 am Beispiel von 5 Zeichen (A bis E).

³⁷Eine weitere Steigerung der eindeutig decodierbaren Codes sind die Präfixcodes, deren Codeworte niemals als Anfangszeichenfolge irgendeines anderen Codewortes vorkommen.

KONZEPTIONIERUNG EINES DRAHTLOSEN EKG-MONITORS

6. EKG-Signalkompression

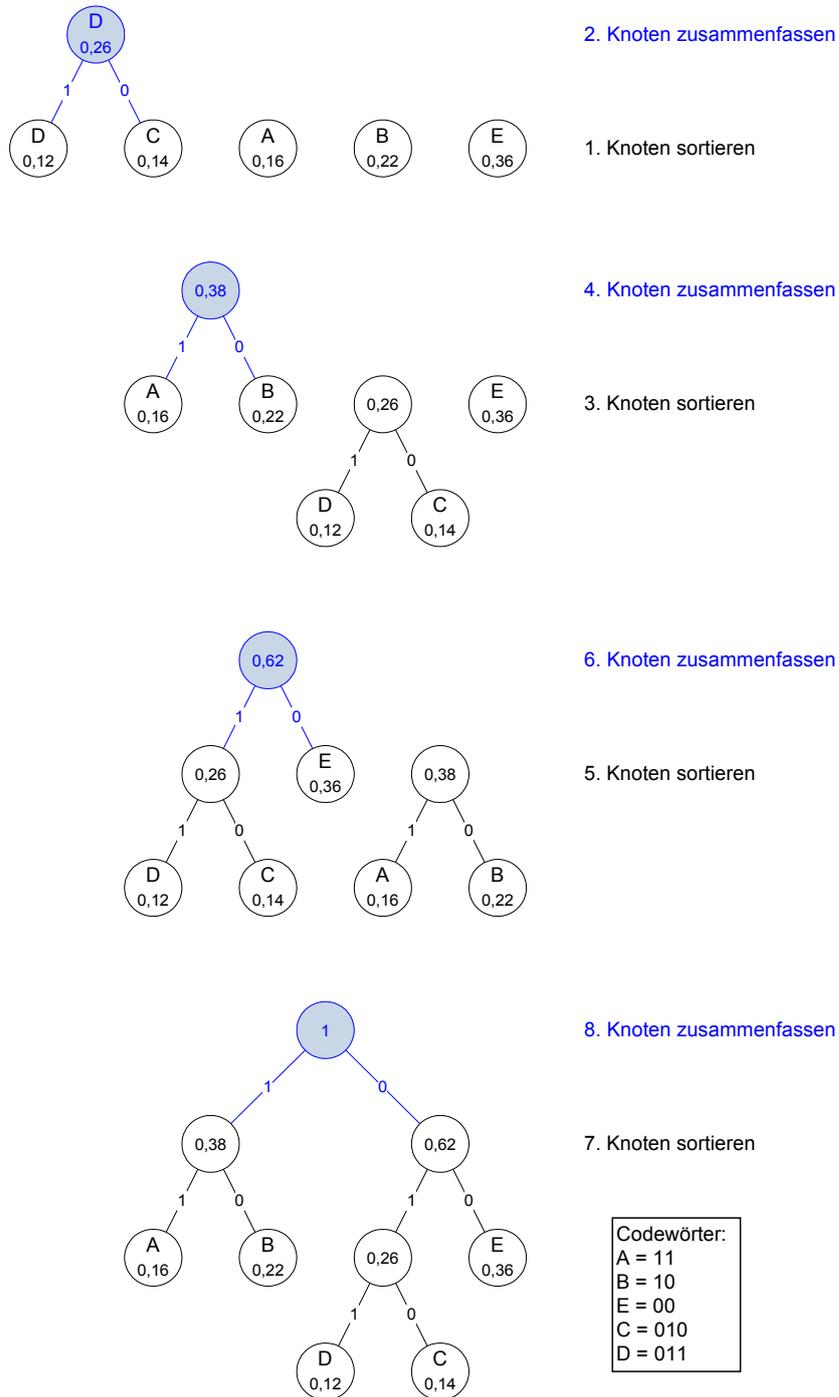


Abbildung 6.7.: Schritte zur Erzeugung eines Huffman-Codebaumes

Der Schlüssel zur Generierung des Huffman-Codes ist die Generierung eines Binärbaumes. Die Blätter enthalten die ursprünglichen Zeichen und die Zweige stehen für die generierten binären Codestellen. Als erster Schritt werden die 5 Zeichen mit ihrer Auftretswahr-

6. EKG-Signalkompression

scheinlichkeit sortiert. Im zweiten Schritt folgt die Zusammenfassung der 2 Zeichen mit der kleinsten Wahrscheinlichkeit zu einem Elternknoten. Der so entstandene Elternknoten wird mit den übrigen Knoten bzw. Blättern neu einsortiert (Schritt 3). Der Schritt 4 gleicht dem Schritt 2 dadurch, dass wieder die 2 Knoten mit der kleinsten Auftretenswahrscheinlichkeit zusammengefasst werden. Alle ungeraden Schritte sortieren die Knoten und die geraden Schritte fassen die Knoten mit der kleinsten Wahrscheinlichkeit zusammen. Man erhält im Schritt 8 den vollständigen Binärbaum. Die so entstandenen Codewörter der Zeichen A bis E im Beispiel sind in der Abbildung 6.7 unten rechts dargestellt.

Setzt man das eben beschriebene Verfahren für die Entropiecodierung des quantisierten Prädiktionsfehlers ein, erhält man die Codewörter in Abbildung 6.8.

Integer	Length	Codeword
32	9	010011000
30	9	010010111
28	8	01100110
26	9	010010110
24	9	010010101
22	6	010000
20	7	0000111
18	7	0000110
16	5	01101
14	5	00101
12	5	00000
10	5	11000
8	4	0111
6	4	1000
4	4	0011
2	4	1010
0	3	111
-2	5	00010
-4	4	1101
-6	4	0101
-8	4	1011
-10	4	1001
-12	5	00100
-14	6	000010
-16	7	0001101
-18	7	1100111
-20	7	0001111
-22	7	0001110
-24	7	1100110
-26	7	0001100
-28	9	010010100
-30	9	010010011
-32	8	01100011

Abbildung 6.8.: Anzeige der Code-Tabelle des Simulators

Man erkennt, dass die Codewortlängen (in Spalte 2) der Häufigkeitsverteilung in Abbildung 6.6 folgen. Der dabei entstandene Codebaum dient auch zur Decodierung bzw. Rückgewinnung der Originaldaten. Der Binärbaum muss dazu auch dem Empfänger bekannt sein. Da einmal festgelegt, kann der Codebaum entweder fest im Speicher vorgehalten werden, oder er wird bei beim Programmstart des Empfängers neu aufgebaut.

6.3. Blockschaltbild

Bisher wurde die Dekorrelation und Entropiecodierung als Methoden zur Datenkompression besprochen. Eine weitere Möglichkeit, die Daten zu reduzieren, ist bereits in Kapitel 3.4 besprochen worden. Es handelt sich hierbei um die Unterabtastung. Bei einer Unterabtastung von $M=4$ wird jeder vierte Signalwert hergenommen. Wichtig in diesem Zusammenhang ist die Berücksichtigung der Abtastrate. Man erhält als neue Abtastrate nach (3.4) einen Wert von $f_{sn} = \frac{256Hz}{4} = 64Hz$. Ein entsprechendes Antialiasing Filter existiert bereits in der Signalaufbereitung im Kapitel 5.5.1 mit einer Grenzfrequenz von $f_a = 31Hz$. Somit ist die Bedingung für die Grenzfrequenz f_a eines Antialiasing Filters $f_a < \frac{1}{2}f_{sn}$ erfüllt. Die Unterabtastungseinheit D ist in Blockschaltbild 6.9 zu sehen.

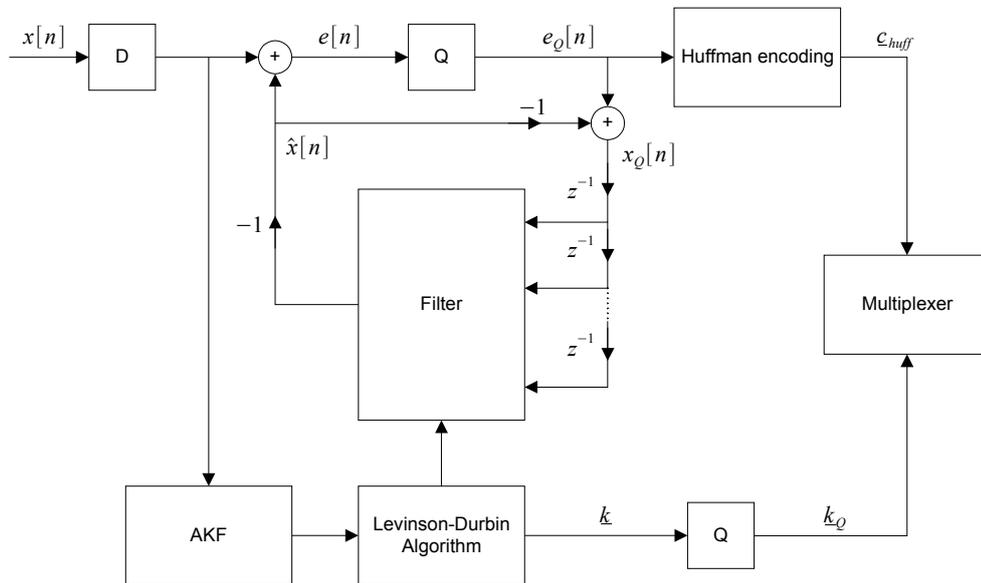


Abbildung 6.9.: Blockschaltbild der Datenkompression

Als Eingang $x[n]$ der Datenkompression kommen die EKG-Signale mit einer 8Bit Quantisierung bei einer Samplingrate von 256Hz an. Durch die Prädiktion mit Quantisierung entsteht ein Prädiktionsfehler $e_Q[n]$, welcher der der Huffman-Codierung zugeführt wird. Das Ergebnis der Huffman-Codierung ist der Vektor \mathcal{C}_{huff} von Huffman-Codes. Zur Signalrückgewinnung auf der Empfängerseite müssen die Filterkoeffizienten der Prädiktion mit übertragen werden. Der Levinson-Durbin-Algorithmus berechnet dazu Reflexionskoeffizienten \underline{k} , die einem Lattice-Filter im Empfänger zugeführt werden (vgl. Kapitel 8.3). Die Reflexionskoeffizienten haben den Vorteil, dass sie die Bedingung $|k| < 1$ erfüllen. Eine nachfolgende 8Bit Quantisierung (Abbildung 6.9 unten) kann effektiv eingesetzt werden. Der so entstandene quantisierte Koeffizienten-Vektor \underline{k}_Q zusammen mit dem Code-Vektor \mathcal{C}_{huff} bilden die Ausgangsdaten der Kompression. Die Serialisierung der Daten zum Er-

6. EKG-Signalkompression

zeugen eines Datenstroms übernimmt der Multiplexer.

Betrachtet man den Grad der Kompression, so kann man feststellen, dass mittels Dekorrelation und anschließender Entropiecodierung eine mittlere Bitlänge L von 4,85Bit/Zeichen (nach (6.5)) entsteht. Mittels Unterabtastung werden die Anzahl der Werte um 4 reduziert. Die Daten werden also um $1 - \frac{4,85\text{Bit}}{4,8\text{Bit}} = 0,85 \rightarrow 85\%$ reduziert.

7. Datenübertragung

Dieses Kapitel behandelt die Datenübertragung des EKG-Signals zwischen dem Monitor und der Basisstation. Ein in diesem Kapitel entwickelter linearer Blockcode sorgt für die Fehlererkennung im Falle eines gestörten Signals. Zur optimalen Bestimmung des Blockcodes folgt eine Betrachtung der Bitfehlerwahrscheinlichkeit und der daraus resultierenden Kanalmatrix. Aus der Codedefinition kann dann das Datenprotokoll definiert werden. Im Detail wird auf die Berechnung des Blockcodes eingegangen unter Berücksichtigung der Fähigkeit, Bündelfehler zu erkennen. Zur Decodierung des Blockcodes wird die Syndromdecodierung eingesetzt. Ein Simulator zur Datenübertragung testet den Blockcode auf seine Funktionalität.

7.1. Bitfehlerwahrscheinlichkeit

Bevor der Blockcode zur Kanalcodierung bestimmt werden kann, ist eine Betrachtung der Bitfehlerwahrscheinlichkeit notwendig. Hierbei wird ein handelsüblicher Empfänger, wie bereits unter 2.2.2 erwähnt, hergenommen. Dieser Empfänger besitzt eine Empfindlichkeit von $P_e = -111dBm$. Es wird im folgenden der Fall betrachtet, bei welchem das Empfangssignal dieser Empfindlichkeit P_e entspricht. Aus [Werner 2006] folgt dabei für das Signal-Rausch-Verhältnis

$$SNR = 10 \cdot \log \frac{P_S}{P_N} = \frac{10^{\frac{P_e}{10}} \cdot 10^{-3}}{k \cdot b \cdot (T_0 + T_z)} = 4,5dB \quad (7.1)$$

bei einer Bandbreite b des Empfängers von 600kHz, bei einer Raumtemperatur T_0 und einer Empfängertemperatur T_z von 40K³⁸.

Damit auf die Bitfehlerwahrscheinlichkeit

$$p_b = \frac{1}{2} \cdot \operatorname{erfc} \left(\sqrt{\frac{1 \cdot E_b}{2 \cdot N_0}} \right) \quad (7.2)$$

³⁸Die Empfängertemperatur liegt hierbei 40K über der Raumtemperatur von 300K. Die Konstante k entspricht der Boltzmann Konstante von $k = 1,38 \cdot 10^{-23} \frac{Ws}{K}$.

7. Datenübertragung

bei binärer Amplitudenumtastung (ASK) bzw. bei binärer Frequenzumtastung (FSK) nach [Roppel 2006] geschlossen werden kann, muss das Verhältnis $\frac{E_b}{N_0}$ aus SNR nach (7.1) bestimmt werden. Dies ist $\frac{E_b}{N_0} = \frac{SNR}{2}$, also 2,25dB. Man erhält in Abbildung 7.1 für ASK bzw. FSK folgende Abhängigkeit zwischen Bitfehlerwahrscheinlichkeit und $\frac{E_b}{N_0}$ -Verhältnis.

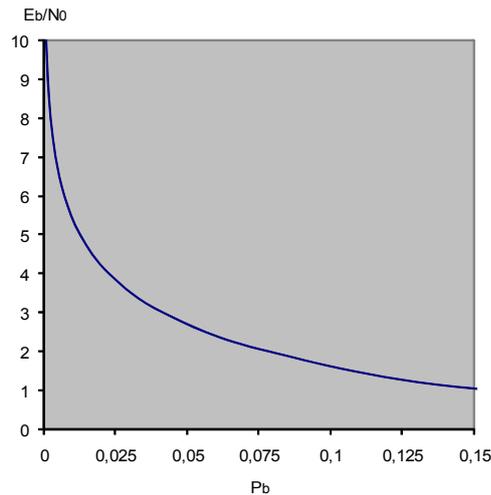


Abbildung 7.1.: Bitfehlerwahrscheinlichkeiten bipolarer und unipolarer Übertragung

Bei genauer Berechnung ergibt sich ein Wert für die Bitfehlerwahrscheinlichkeit von $p_b = 0,1175$.

7.2. Kanalmatrix

Bei der Übertragung über einen symmetrischen Binärkanal (BSC) resultiert aus dem Eingangsbit $x_1 = 1$ das Ausgangsbit $y_1 = 1$ mit einer Wahrscheinlichkeit $1 - p = 1 - p_b = 1 - 0,1175 = 0,8825$. Der zugehörige Graph zeigt Abbildung 7.2. Analog gilt dazu für das Eingangsbit $x_2 = 0$ die gleiche Wahrscheinlichkeit für das Ausgangsbit $y_2 = 0$.

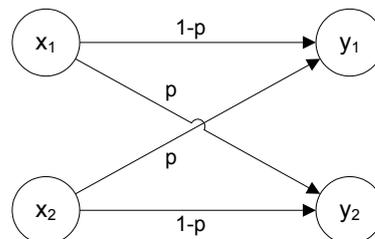


Abbildung 7.2.: Kanalmatrix des binären symmetrischen Kanals

Aus diesem Sachverhalt kann nun die Kanalmatrix für den BSC bestimmt werden:

7. Datenübertragung

$$p(y|x) = \begin{pmatrix} 1 - p_b & p_b \\ p_b & 1 - p_b \end{pmatrix} = \begin{pmatrix} 0,8825 & 0,1175 \\ 0,1175 & 0,8825 \end{pmatrix} \quad (7.3)$$

Aus der Kanalmatrix lässt sich nun die maximal mögliche Kanalkapazität C bestimmen. Die Kanalkapazität gibt die Information an, die maximal pro Zeichen übertragen werden kann. Beim symmetrischen Kanal wird die Kanalkapazität C wie folgt bestimmt:

$$C = H_{max}(y) - H(y|x) = H_{max}(y) - \left[- \sum_{j=1}^q p(x_j) \sum_{i=1}^q p(y_i|x_j) \text{ld} p(y_i|x_j) \right] \quad (7.4)$$

Beim symmetrischen Binärkanal kann diese nach [Carl 2009] folgendermassen

$$C = (1 + ((1 - p_b) \text{ld}(1 - p_b) + p_b \text{ld} p_b)) \quad (7.5)$$

vereinfacht werden. Man erhält für die Kanalkapazität einen Wert von C = 0,478 Bit/Zeichen.

Nach dem Kanalcodierungstheorem gilt, dass über einen diskreten, gedächtnislosen Kanal mit der Kanalkapazität C Übertragungen mit Coderaten $R < C$ mit beliebig kleiner Fehlerwahrscheinlichkeit möglich sind. Die Codrate R ist definiert mit

$$R = \frac{n}{m} \quad (7.6)$$

für gleiche Anzahl von Eingangs- und Ausgangssymbolen, wobei n für das n-stellige Informationswort und m für das m-stellige Codewort steht. Als weitere Randbedingung soll gelten, dass maximal 8-stellige Codeworte übertragen werden sollen, um auch die Einbettung in 8Bit Datenprotokollen zu ermöglichen. Für das n-stellige Informationswort kann man nun die Bedingung $n < C \cdot m$ schreiben. Somit ergibt sich für $n_{max} = 0,478 \cdot 8 = 3,8$ stellen. Es wird n=3 festgelegt. Die Kurzbeschreibung des Blockcodes lautet:

$$(8, 3) - \text{Blockcode} \quad (7.7)$$

Für die Abschätzung der Restfehlerwahrscheinlichkeit des fehlererkennenden Codes kann für einen besseren Kanal mit $p < 0,5$ die Formel

$$p_{Rdet} = \frac{1}{2^{m-n}} \quad (7.8)$$

7. Datenübertragung

aufgestellt werden³⁹. Man erhält für die Restfehlerwahrscheinlichkeit einen Wert von $p_{Rdet} = 0,03125$.

7.3. Blockcode-Generierung

In Kapitel 3.5 wurde bereits die Fehlererkennungsfähigkeit des CRC-Codes bezüglich Bündelfehlern erwähnt. Die Wahrscheinlichkeit p_{RdetBF} , dass ein Bündelfehler der Länge t bei $k = m-n$ Schutzbits unerkannt bleibt, zeigt folgende Gleichung aus [Carl 2009]:

$$p_{RdetBF} = \begin{cases} 0 & t < k-1 \\ 2^{-k+1} & t = k+1 \\ 2^{-k} & t > k+1 \end{cases} \quad (7.9)$$

Häufig werden CRC-Codes mit einem Generatorpolynom erzeugt, das aus einem Generatorpolynom $g_{prim}(x)$ und dem primitiven Polynom 1.Ordnung $(x+1)$ besteht.

$$g(x) = g_{prim}(x) \cdot (x + 1) \quad (7.10)$$

Diese Art der Generierung erhöht die Minimaldistanz des Codes auf $d_{min} = 4$. Das bedeutet die Erkennung von 3 Einzelfehlern ($e_{det=3}$). Die Kurzbezeichnung des Blockcodes kann um den Wert für d_{min} erweitert werden (3.6) und lautet demnach (8,3,4). Für das unerkannt bleiben von Bündelfehlern nach Gleichung (7.9) mit dem (8,3,4)-Blockcode ($k=5$), erhält man die entsprechenden Wahrscheinlichkeiten in Abhängigkeit der Bündelfehlerlänge von

- $p_{RdetBF}|_{t=4..5} = 0$
- $p_{RdetBF}|_{t=6} = 0,0625$
- $p_{RdetBF}|_{t>6} = 0,03125$

Zur Überprüfung, ob der (8,3,4)-Blockcode gültig ist, müssen die Singleton-, Hamming- und die Plotkin-Schranken⁴⁰ erfüllt sein. Die Singleton-Schranke

³⁹Die Restfehlerwahrscheinlichkeit p_{Rdet} für einen fehlererkennenden Code gibt die Wahrscheinlichkeit für ein Empfangswort an, das bei einer gestörten Übertragung wieder ein Codewort ist. Dieses Codewort entspricht aber nicht dem gesendeten Codewort.

⁴⁰Die Gilbert-Varshamov-Schranke stellt eine hinreichende Bedingung dar und muss somit nicht erfüllt sein.

7. Datenübertragung

$$m - n \geq d_{min} - 1 \quad (7.11)$$

ist erfüllt, da $5 \geq 3$ ist. Die Hamming-Schranke

$$m - n \geq ld \left[\sum_{i=0}^{e_{kor}} \binom{m}{i} \right] \quad (7.12)$$

mit

$$e_{kor} = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor = \left\lfloor \frac{3}{2} \right\rfloor = 1$$

$$5 \geq ld \left[\sum_{i=0}^1 \binom{8}{i} \right] = ld \left[\binom{8}{0} + \binom{8}{1} \right] = ld(1 + 8) = 3, 17$$

ist auch erfüllt. Die Plotkin-Schranke

$$d_{min} \leq \frac{m}{2} \cdot \frac{2^n}{2^n - 1} \quad (7.13)$$

$$4 \leq 4 \cdot \frac{8}{7} = 4,57$$

ist ebenso erfüllt.

Bei der Auswahl des Generatorpolynoms $g_{prim}(x)$ für (7.10) der Ordnung $k=m-n-1=4$ erhält man nach [Carl 2009] das Generatorpolynom $g_{prim}(x) = x^4 + x + 1$. Da $m < z^{k-1} - 1$ erfüllt ist, spricht man auch von einem verkürzten zyklischen Code⁴¹. Das Generatorpolynom setzt sich somit aus (7.10)

$$g(x) = (x^4 + x + 1) \cdot (x + 1) = x^5 + x^4 + x^2 + 1 \quad (7.14)$$

zusammen. Um einen systematischen Code zu erhalten, berechnen sich die Codeworte $c(x)$ in Abhängigkeit der Infoworte $i(x)$ und des Generatorpolynoms $g(x)$ folgendermassen:

⁴¹Diese werden in der Praxis häufig eingesetzt, da dadurch eine effiziente Implementierbarkeit der erforderlichen Polynomdivision möglich ist.

7. Datenübertragung

$$\begin{aligned}
 c(x) &= i(x) \cdot x^{m-n} + i(x) \cdot x^{m-n} \bmod g(x) \\
 c(x) &= i(x) \cdot x^5 + i(x) \cdot x^5 \bmod x^5 + x^4 + x^2 + 1
 \end{aligned}
 \tag{7.15}$$

Die dabei entstehenden Codeworte werden in folgender Tabelle dargestellt. In der linken und mittleren Spalte stehen die Infoworte und die Codeworte in der Polynomschreibweise, in der rechten Spalte stehen die Codeworte in Hexadezimal-Darstellung.

Tabelle 7.1.: Codeworte in Polynom-Darstellung und in hexadezimaler Schreibweise

Infoworte	Codewort-Polynome	Codeworte hexadezimal
0	0	0x00
1	$x^5 + x^4 + x^2 + 1$	0x35
x	$x^6 + x^4 + x^3 + x^2 + x + 1$	0x5F
$x + 1$	$x^6 + x^5 + x^3 + x$	0x6A
x^2	$x^7 + x^3 + x + 1$	0x8B
$x^2 + 1$	$x^7 + x^5 + x^4 + x^3 + x^2 + x$	0xBE
$x^2 + x$	$x^7 + x^6 + x^4 + x^2$	0xD4
$x^2 + x + 1$	$x^7 + x^6 + x^5 + 1$	0xE1

7.4. Datenprotokoll

Aus den Erkenntnissen aus dem vorherigem Kapitel bezüglich des (8,3,4)-Blockcodes, kann nun das Datenprotokoll zur Übertragung des EKG-Signals zwischen Sender und Empfänger in Abbildung 7.4 festgelegt werden.

(8,3,4)-Blockcode											
8Bit	8Bit
Coefficient 1			Coefficient n			Code 1			Code m		
Predictor Coefficients						Huffman Codes					

Abbildung 7.4: Datenprotokoll der Übertragung

Die Übertragung besteht aus dem (8,3,4)-Blockcode. Zuerst werden die Prädiktor Koeffizienten übertragen, danach folgen die Huffman-Codes des komprimierten EKG-Signals. Vernachlässigt man die Anzahl der Koeffizienten, so besteht ein EKG-Datenblock aus 512 Huffman-Codes. Der Huffman-Code mit der größten Länge m_i beträgt $m_i|_{max} = 9$. Somit sind maximal 3 Blockcodes für ein Huffman-Codewort notwendig. Das entspricht eine Bitlänge L_B von $L_B = 512 \cdot 3 \cdot 8Bit = 12288Bit$. Ein Datenblock besteht vor der Kompression aus $n=2048$ Samples mit einer Samplingrate von $t_n = \frac{1}{256Hz}$. Die Aufzeichnungsdauer T_R berechnet sich nun aus $T_R = n \cdot t_n = 2048 \cdot \frac{1}{256Hz}$ und ergibt $T_R = 8s$. Das heisst, es fallen 12288Bit in einem Zeitraum von 8 Sekunden an, bzw. anders formuliert, es kommt zu einer Bitrate von $r = 1536 \frac{Bit}{s}$.

Der erwähnte Empfänger in 2.2.2 besitzt eine NF-Bandbreite⁴² von $b_{NF} = 30kHz$. Mit der bereits angesprochenen BiPhase-Mark-Leitungscodierung (vgl. Kapitel 2.2.2) ist somit eine maximale Bitrate $r_{BiPhase}$ von $r_{BiPhase}|_{max,1} = 15000 \frac{Bit}{s}$ möglich. Die dabei auftretende kleinste Bitrate⁴³ beträgt $r_{BiPhase}|_{max,0} = \frac{15000}{2} \frac{Bit}{s} = 7500 \frac{Bit}{s}$. Mit dieser Worst-Case-Betrachtung ist die Bedingung $r < r_{BiPhase}|_{max,0}$ erfüllt. In Worten ausgedrückt: Die notwendige Bitrate r ist für den (8,3,4)-Blockcode kleiner als die Bitrate, die der Empfänger maximal für die Übertragung von Nullen aufbringen kann.

⁴²Diese Angabe bezieht sich auf ein Rechtecksignal mit der maximalen Frequenz von 30kHz.

⁴³Für den rein theoretischen Fall, dass nur gleiche Bits mit dem Wert 0 übertragen werden.

7.5. Blockcode-Decodierung

Für eine einfache Realisierung der Decodierung kommt die Syndromdecodierung in Frage. Der Decoder berechnet hierbei das Syndrom s aus dem Empfangswort r . Bezogen auf das Syndrom s kann ein Übertragungsfehler detektiert werden, wenn $s \neq 0$ gilt. Um das Syndrom berechnen zu können, muss zunächst die Generatormatrix G bestimmt werden:

$$G = \begin{bmatrix} g(x) \cdot x^2 \\ g(x) \cdot x \\ g(x) \end{bmatrix} = \begin{bmatrix} x^7 + x^6 + x^4 + x^2 \\ x^6 + x^5 + x^3 + x \\ x^5 + x^4 + x^2 + 1 \end{bmatrix} = \begin{bmatrix} 11010100 \\ 01101010 \\ 00110101 \end{bmatrix} \quad (7.16)$$

Die Generatormatrix ist nicht-systematisch. Die klar-systematische Generatormatrix G_{ks} besteht aus einer $n \times n$ - Einheitsmatrix E_n und einer $n \times (m-n)$ - Prüfstellenmatrix P . Durch elementare Operationen [Carl 2009] kann nun die nicht-systematische Generatormatrix in eine klar-systematische Generatormatrix überführt werden.

$$G_{ks} = \left[E_n \mid P \right] = \left[\begin{array}{c|ccc} 100 & 01011 \\ 010 & 11111 \\ 001 & 10101 \end{array} \right] \quad (7.17)$$

Die Prüfmatrix H kann nun aus

$$H = \left[P^T \mid E_{m-n} \right] = \left[P^T \mid E_5 \right] = \left[\begin{array}{c|cccc} 011 & 10000 \\ 110 & 01000 \\ 011 & 00100 \\ 110 & 00010 \\ 111 & 00001 \end{array} \right] \quad (7.18)$$

bestimmt werden. Das Syndrom s berechnet sich aus dem Empfangscodewort r und der transformierten Prüfmatrix H^T folgendermassen:

$$s = r \odot H^T = r \odot \left[P^T \middle| E_5 \right]^T$$

$$s = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7 \ r_8] \odot \begin{bmatrix} 01011 \\ 11111 \\ 10101 \\ 10000 \\ 01000 \\ 00100 \\ 00010 \\ 00001 \end{bmatrix} = [s_1 \ s_2 \ s_3 \ s_4 \ s_5] \quad (7.19)$$

Im Falle einer fehlerfreien Übertragung erhält man für das Syndrom s einen Null-Vektor $s = [00000]$. Der Decoder muss sich lediglich die Prüfmatrix H^T merken und kann somit die fehlerhaften Empfangscodeworte r detektieren.

7.6. Simulator

Zur Simulation der Datenübertragung, wurde hierzu ein Simulator in C++ mit Hilfe der Qt-Bibliothek entwickelt. Das Programm ist unter Windows lauffähig, kann aber aufgrund der Plattformunabhängigkeit von Qt auch auf anderen Plattformen wie z.B. Unix übersetzt werden. Das Klassendiagramm der Implementierung zeigt Abbildung 7.3.

Die grau hinterlegten Klassen *model*, *controller* und *MainWindow* stellen das MVC-Pattern⁴⁴ dar. Dieses Entwurfsmuster (Pattern) stellt die Grundstruktur für die Applikation bereit. Somit lassen sich die Programme übersichtlicher gestalten und es gibt konkrete Verantwortlichkeiten der Klassen. Der *controller*, wie auch der Name schon vermuten lässt, übernimmt die Kontrolle. In ihm steckt somit die Ablaufsteuerung. Die *model*-Klasse beinhaltet alle Daten, die die Applikation benötigt. Die View-Klasse, hier als *MainWindow* bezeichnet implementiert die Schnittstelle zum Benutzer. Damit nun Zirkelbezüge⁴⁵ vermieden werden können, kommt eines der Stärken von Qt zum Einsatz: Das Signal-Slot-Konzept. Damit kann man eine Art „Callback“ definieren, mit dem der Sender ein Signal aussendet und der Empfänger (Slot) das Signal empfängt. Im Klassendiagramm sind die Callbacks

⁴⁴Das Model-View-Controller-Muster wird häufig in Entwicklung von GUI Applikationen verwendet.

⁴⁵Zirkelbezüge entstehen, wenn die Abhängigkeiten der Klassen einen „Kreis“ bilden. Bezogen auf die Model-View-Controller-Klassen hätte man beispielsweise eine Abhängigkeitskette von Controller nach View, von View nach Model und von Model wieder zurück zum Controller. Solch eine Struktur ist sehr fehleranfällig und schlecht wartbar.

7. Datenübertragung

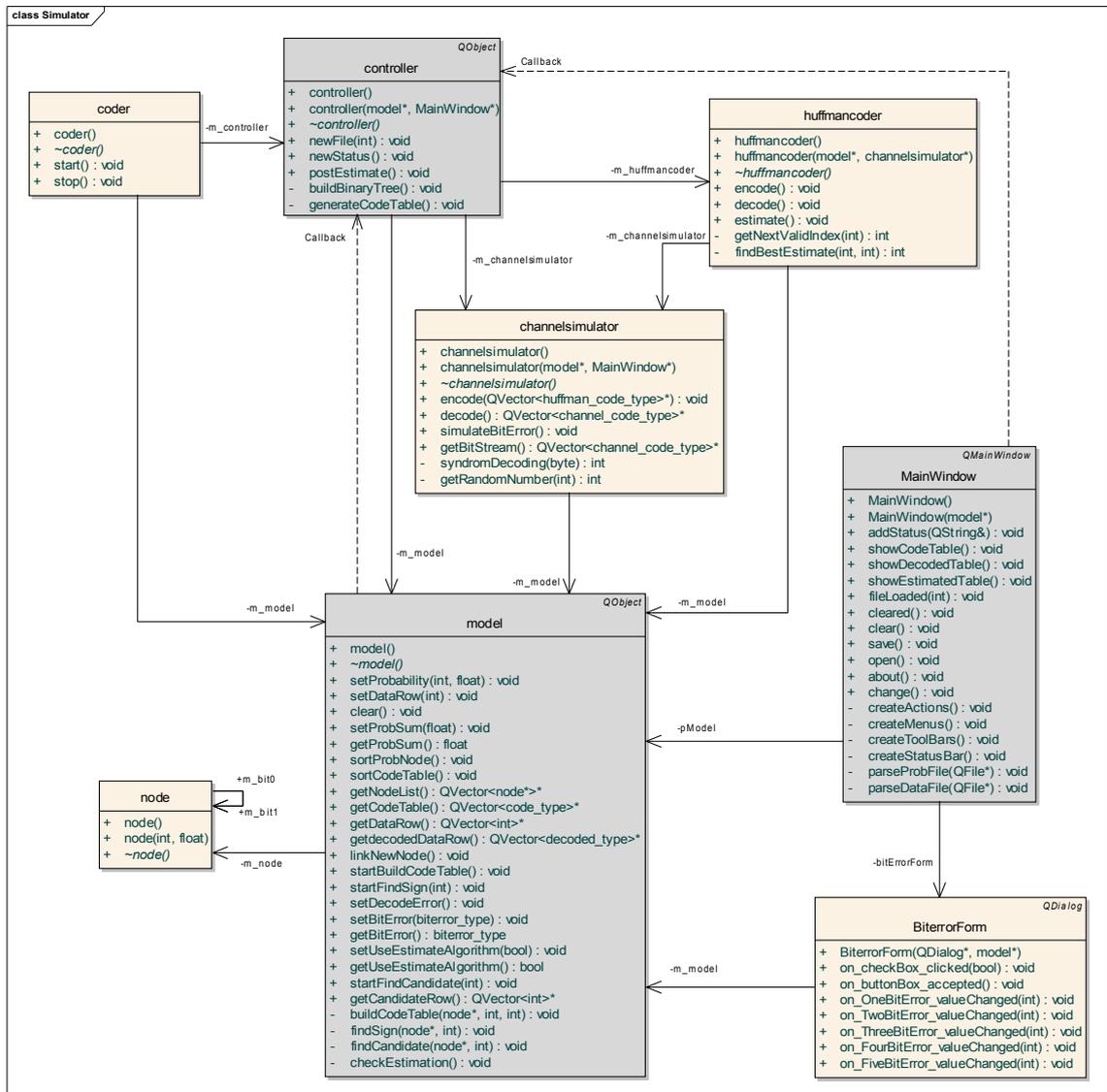


Abbildung 7.3.: Klassendiagramm des Simulators

als gestrichelte Pfeile gekennzeichnet, da sie keine Klassenabhängigkeiten bilden.

Die Klasse *coder* dient als Fabrik und erzeugt die Klassen des MVC-Patterns. Darüber hinaus startet und stoppt sie den Simulator.

Die Klasse *huffmancoder* codiert das Eingangssignal mit dem Huffman-Code und stellt den Datenstrom der Klasse *channelsimulator* zur Verfügung. Nach der simulierten Übertragung wird das Signal wieder zurück codiert und am Bildschirm in Form einer Tabelle angezeigt.

Der *channelsimulator* übernimmt die Codierung und Decodierung des Blockcodes und simuliert Übertragungsfehler. Die Anzahl der Bit-Fehler kann über die Eingabemaske *Bi-*

7. Datenübertragung

terrorForm eingegeben werden.

Die *node*-Klasse stellt einen Knoten im Binärbaum für die Huffman-Codierung dar. Die Erzeugung des Binärbaumes wurde im Kapitel 6.2.2 bereits besprochen. Eine Beschreibung der Bedienoberfläche des Simulators befindet sich im Anhang A.7. Der Sourcecode des Simulators befindet sich auf der CD zur Masterarbeit unter */Qt*.

7.7. Resümee

In diesem Kapitel wurde der (8,3,4)-Blockcode zur Kanalcodierung ermittelt. Dieser entspricht einem verkürzten zyklischen Blockcode mit der Fähigkeit der Bündelfehlererkennung. Die dabei resultierende Hammingdistanz erhöht sich dadurch auf $d_{min} = 4$ und erlaubt somit eine Fehlererkennung von $e_{det} = 3$ Einzelfehlern. Basierend auf diesem Code werden die Prädiktorkoeffizienten und das huffmancodierte Prädiktionsfehlersignal übertragen. Zur Simulation der Datenübertragung wurde ein Simulator entwickelt. Dieser erzeugt Übertragungsfehler und kann somit die Wirksamkeit des Blockcodes überprüfen.

8. Signalmrückgewinnung

Nachdem in den vorigen Kapiteln die Datenkompression im Fordergrund stand, beschäftigt sich dieses Kapitel mit der Wiederherstellung des ursprünglichen Signals vor der Kompression. Mit Rückgewinnung ist also die Wiederherstellung des EKG-Signals aus dem Prädiktionsfehler bezeichnet. Dazu zeigt der erste Abschnitt das dazu notwendige Blockschaltbild. Die Huffman-Decodierung und die Prädiktion-Decodierung bilden hierbei die wichtigsten funktionalen Einheiten. Diese werden in den folgenden Abschnitten detailliert beschrieben.

8.1. Blockschaftbild

Nach der Signalaufbereitung in Abbildung 3.1 wird das EKG-Signal komprimiert, um eine Reduktion der Daten zu erreichen. Danach folgt die Datenübertragung zum Empfänger, der nun das EKG-Signal wieder herstellt. Das Blockschaftbild in Abbildung 8.1 zeigt dazu die notwendigen funktionalen Einheiten. Die Einbettung der Signalaufbereitung erfolgt im Empfänger, bzw. im Webserver nach Abbildung 2.1.

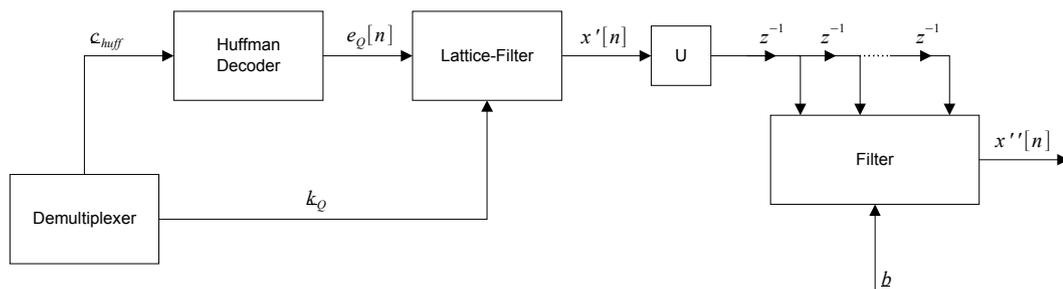


Abbildung 8.1.: Blockschaftbild zur Signalmrückgewinnung

Nach der Datenübertragung sorgt der Demultiplexer zur Aufspaltung der Huffman-Codes c_{huff} und der Koeffizienten k_Q . Im Block Huffman-Decoder müssen die Huffman-Codes wieder decodiert werden. Man erhält daraus den quantisierten Prädiktionsfehler $e_Q[n]$. Das Lattice-Filter hat die Aufgabe, den quantisierten Prädiktionsfehler wieder in das EKG-

8. Signalmrückgewinnung

Signal zurück zu wandeln. Dazu benötigt es die quantisierten Reflexionskoeffizienten⁴⁶ k_Q . Das Ausgangssignal $x'[n]$ des Lattice-Filters ist das zurückgewonnene EKG-Signal, aber noch um den Faktor 4 herunter getaktet. Die folgenden Blöcke U und Filter bilden zusammen ein Interpolationsfilter. Dieses Filter taktet das Signal um den Faktor 4 wieder hoch und interpoliert die durch das Hochtakten erzeugten Nullstellen zu einem „kontinuierlichen“ Signal⁴⁷. Das Interpolationsfilter ist nicht zwingend notwendig. Es dient hier lediglich dazu, das EKG-Signal wieder auf die ursprüngliche Samplingfrequenz von 256Hz hoch zu takten.

8.2. Huffman-Decoder

Für die Codegenerierung der Huffman-Codes in Kapitel 6.2.2 wurde ein Binärbaum erzeugt. Die Blätter des Baumes stellen die zu codierenden Zeichen dar. Die Zweige bezeichnen eine Binärstelle des Huffman-Codes (vgl. Abbildung 6.7). Der erzeugte Binärbaum dient nun auch dazu die Codes im Empfänger wieder zu decodieren. In der Abbildung 8.2 wird der bereits beispielhaft generierte Binärbaum verwendet, um die Decodierung des Zeichens „C“ zu beschreiben.

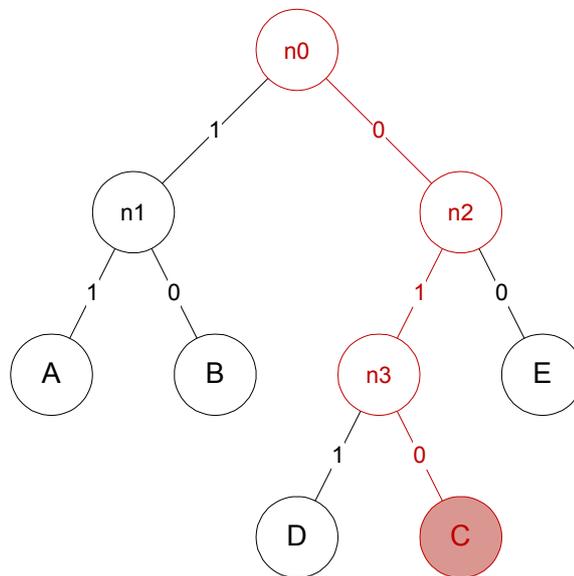


Abbildung 8.2.: Binärbaum zur Decodierung der Huffman-Codes

⁴⁶In Abbildung 6.9 werden die Reflexionskoeffizienten für ein Lattice-Filter aus dem Levinson-Durbin Algorithmus erzeugt.

⁴⁷Durch das Hochtakten (upsampling) von 4 entstehen 3 neue Stützstellen, die den Wert 0 haben. Durch das anschließende Filter werden die 3 neuen Werte dem ursprünglichen Signal vor dem Heruntertakten angenähert.

8. Signalrückgewinnung

Die Knoten (nodes) des Baumes sind mit n0 bis n3 durchnummeriert. Der entsprechende Huffman-Code für das Zeichen „C“ lautet: „010“. Vom Startknoten n0 aus führt die erste Null über den rechten Zweig zum Knoten n2. Die 1 führt zum Knoten n3 und die letzte Null schliesslich zum Blatt mit dem Zeichen „C“.

Betrachtet man die Knoten als Zustände und die Zweige als Zustandsübergänge, so kann man den Binärbaum auch als Zustandsdiagramm, wie in Abbildung 8.3 dargestellt, zeichnen.

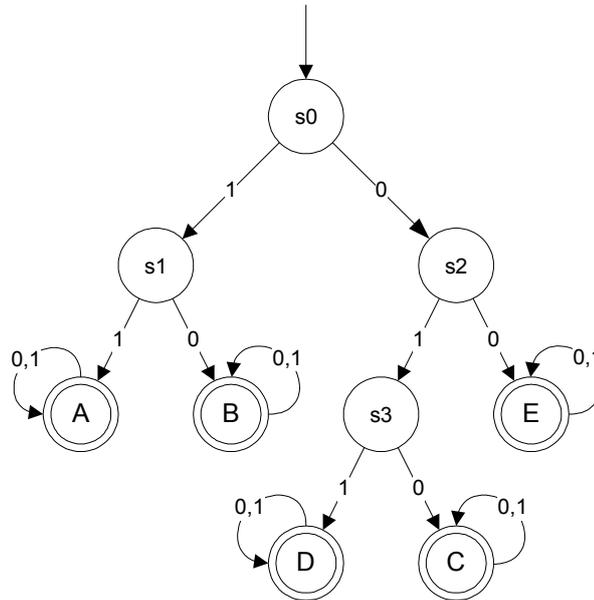


Abbildung 8.3.: Endlicher deterministischer Automat zur Decodierung der Huffman-Codes

Man erhält einen endlichen deterministischen Automaten⁴⁸ mit der Sprache L_A .

$$L_A = \{w \in X^* | \delta_{(s_0,w)}^* \in F\} \tag{8.1}$$

In Worten bedeutet die Gleichung (8.1), dass für jedes Wort w aus der Menge $X=\{0,1\}$ Zustandsübergänge $\delta_{(s_0,w)}^*$ existieren, die zu einem Endezustand F in diesem Beispiel $F=\{A,B,C,D,E\}$ führen. Die Menge der Zustände S beinhaltet $S = \{s_0, s_1, s_2, s_3, A, B, C, D, E\}$, daraus folgt, dass $F \subseteq S$ und somit auch $\delta_{(s_0,w)}^* \subseteq S$ gelten muss.

Überträgt man die oben erwähnten Zusammenhänge auf den endlichen deterministischen Automaten des Huffman-Decoders im Empfänger des EKG-Monitors, so besteht die Menge der Endezustände F aus:

⁴⁸Es existiert auch ein endlicher nicht-deterministischer Automat. Der Unterschied besteht hauptsächlich darin, dass dieser Automat mehrere Anfangszustände und mehrere Folgezustände bei Eingabe eines Zeichens haben kann.

8. Signalmrückgewinnung

$$F = \{ -64, -62, -60, \dots, -2, 0, 2, \dots, 60, 62 \} \quad (8.2)$$

Die Zustandsmenge S fasst alle Zustände $s_x | x \in \mathbb{N}, F$ zusammen:

$$S = (\{s_0, s_1, \dots, s_n\}, F) \quad n \in \mathbb{N} \quad (8.3)$$

Der Automat wurde im Simulator zur Datenübertragung realisiert. Nach erfolgreicher Datenübertragung erscheint ein Übersichtsfenster mit dem Titel „Decoded Table“ (Abbildung 8.4). In diesem Fenster ist in der linken Spalte der quantisierte Prädiktionsfehler vor der Übertragung zu sehen und in der rechten Spalte der quantisierte Prädiktionsfehler nach der Huffman-Decodierung.

	transmitted	received
1	0	0
2	0	0
3	0	0
4	14	14
5	42	42
6	-2	-2
7	-46	-46
8	12	12
9	-4	-4
10	-26	-26
11	4	4
12	12	12
13	14	14
14	18	18
15	12	12
16	2	2
17	4	4

Abbildung 8.4.: Anzeige der decodierten Daten des Simulators

Damit im Automat auch im Fall einer fehlerhaften Übertragung immer ein Endzustand $\delta_{(s_0,w)}^* \in F$ erreicht wird, müssen alle Wörter w eine Mächtigkeit

$$|w| = \max_{i=0, \dots, N-1} \{m_i\}$$

8. Signalmrückgewinnung

aufweisen. Man erhält für $|w| = 9$, da nach Abbildung 6.8 der längste Code eine Bitzahl von $m=9$ aufweist. Für die Übertragung bedeutet dies, dass immer 3 (8,3)-Blockcodes übertragen werden müssen. Bei einer Übertragung eines Codewortes mit $m=6$, werden demnach die letzten 3 Bitstellen mit Nullen aufgefüllt. Ist einmal ein Endzustand erreicht, bleibt der Automat in diesem Zustand, egal wieviele weitere Zeichen kommen (vgl. Abbildung 8.3). Ein neuer Decodier-Vorgang setzt den Automaten in seinen Anfangszustand s_0 zurück.

Im Falle einer fehlerhaften Übertragung mit Bitfehler wird dies durch die Syndromdecodierung erkannt. Bei diesem Umstand werden verschiedene Decodierstrategien im Huffman-Decoder dazu vorgeschlagen:

1. Die fehlerhafte Übertragung wird dem Benutzer im Webserver angezeigt. Der Benutzer startet die Übertragung erneut oder er akzeptiert die fehlenden Stellen im EKG-Signal.
2. Der Huffman-Decoder versucht das fehlerhafte Codewort zu korrigieren, indem er die 8 Möglichkeiten des falschen (8,3)-Blockcodes im Binärbaum durchspielt. Das dabei gefundene kürzeste Codewort wird akzeptiert, da dies das wahrscheinlichere Codewort gemäss Entropiecodierung darstellt.
3. Der Huffman-Decoder versucht das fehlerhafte Codewort zu korrigieren, indem er die 8 Möglichkeiten des falschen (8,3)-Blockcodes im Binärbaum durchspielt. Die dadurch resultierenden Werte werden mit den vorangegangenen Werten aus einer korrekten Übertragung verglichen. Die Schätzung des wahrscheinlichsten Codewortes hängt nun davon ab, mit welcher Prädiktor Ordnung die Übertragung erfolgt. Ist die Ordnung kleiner, so ist das Prädiktionsfehlersignal weniger stark dekorreliert und das Quellengedächtnis ist nicht ganz entfernt. Das bedeutet, die Prädiktionsfehlerwerte haben eine gewisse Abhängigkeit zu ihren Vorgängern. Interpoliert man die fehlerhaften Stellen mit den 8 Möglichkeiten, so ergibt sich das wahrscheinlichste Codewort aus dem geringsten Abstand zur Interpolation.

8.3. Prädiktor-Decodierung

Nach der Huffman-Decodierung liegt nun das quantisierte Prädiktionsfehlersignal $e_Q[n]$ vor. Damit man daraus das EKG-Signal erhält, muss die Prädiktion rückgängig gemacht werden. Dies geschieht mit dem Lattice-Filter zur Prädiktor-Decodierung in Abbildung 8.1. Die notwendigen Koeffizienten k_Q für dieses Filter wurden mit übertragen. Hierbei handelt es sich um Reflexionskoeffizienten, die im Sender durch den Levinson-Durbin-Algorithmus berechnet wurden. Mit diesen Koeffizienten kann ein All-pole-Lattice-Filter

8. Signalmrückgewinnung

zur Prädiktion-Decodierung bestimmt werden [Werner 2008]. Dieses Lattice-Filter zeigt Abbildung 8.5.

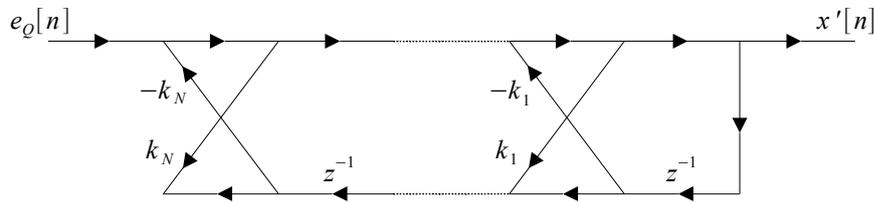


Abbildung 8.5.: Prädiktion Decodierung mittels All-pole-Lattice-Filter (Bildquelle: [Werner 2008] modifiziert)

Die Verwendung von Lattice-Filtern hat Vorteile bezüglich ihrer adaptiven und stufenförmigen Erweiterbarkeit [Werner 2008]. Darüber hinaus sind sie relativ robust gegenüber Wortlängeneffekten und sie besitzen ein einfaches Stabilitätskriterium⁴⁹. Das bedeutet, dass bei der Erhöhung der Prädiktorordnung im Sender der Empfänger in seiner Implementierung nicht geändert werden muss. Werden mehr Koeffizienten k_{N+m} übertragen, verlängert sich die Struktur in Abbildung 8.5 dementsprechend um m-Segmente⁵⁰. Die Übertragungsfunktion $H(z)$ entspricht der eines IIR-Filters

$$H(z) = \frac{1}{A_N(z)} = \frac{1}{\sum_{i=0}^N a_i z^{-i}} \quad (8.4)$$

Das Ausgangssignal $x'[n]$ entspricht dem EKG-Signal vor der Prädiktion bzw. nach dem Downsampling in Abbildung 6.9.

8.4. Interpolation

Nach der Prädiktionsdecodierung erhält man das EKG-Signal, jedoch aufgrund der Datenkompression in Abbildung 6.9 um den Faktor 4 herunter getaktet. Sollte es notwendig sein, das Signal wieder in die ursprüngliche Abtastrate von 256Hz zurück zu konvertieren, muss ein Upsampling⁵¹ durchgeführt werden. Der Funktionsblock „U“ in Abbildung 8.1 stellt diese Funktion zur Verfügung. Hierbei wird zwischen zwei Abtastwerten 3 weitere Stützstellen eingefügt. Diese Stützstellen haben nach der Upsample Funktion den Wert

⁴⁹Das All-pole-System ist strikt stabil, wenn $|k_i| < 1$ für $i=1:N$ gilt.

⁵⁰Mit Segment ist hier ein Verzögerungsglied z^{-1} und die Grafen für die Koeffizienten k_N gemeint.

⁵¹Dies ist in dieser Anwendung nicht zwingend notwendig, soll aber dennoch an dieser Stelle erwähnt werden.

8. Signalrückgewinnung

0. Damit die Werte dem EKG-Signalverlauf „angenähert“ werden können, benötigt man ein Interpolationsfilter. Die Koeffizienten für das FIR-Filter kann man mittels der Matlab Funktion *intfilt* ermitteln. Man erhält die Koeffizienten in Tabelle 8.1 für eine bandbegrenzte Interpolation mit einem Bandbegrenzungsfaktor von 0.5.

Tabelle 8.1.: FIR-Filterkoeffizienten zur Interpolation

Koeffizient	Wert
b_0	-0,05827
b_1	-0,0927
b_2	-0,0794
b_3	0,0000
b_4	0,2837
b_5	0,5861
b_6	0,8444
b_7	1,0000
b_8	0,8444
b_9	0,5861
b_{10}	0,2837
b_{11}	0,0000
b_{12}	-0,0794
b_{13}	-0,0927
b_{14}	-0,0582

Die Koeffizienten können verwendet werden, um ein Interpolationsfilter für diese Anwendung zu realisieren. Das Ausgangssignal des Interpolationsfilters wird in Abbildung 8.1 mit $x''[n]$ bezeichnet.

8.5. Resümee

In diesem Kapitel wurde gezeigt, wie aus den Huffman-Codes das EKG-Signal zurück gewonnen werden kann. Optional ist es mit dem Upsampling und einem Interpolationsfilter möglich, das EKG-Signal in seine ursprüngliche Abtastrate von 256Hz zu konvertieren. Stellt man das Spektrum des EKG-Signals vor der Datenkompression im Sender mit dem Spektrum des EKG-Signal in der Signalrückgewinnung mit Upsampling und Interpolation gegenüber, erhält man die Spektren wie in Abbildung 8.6 dargestellt.

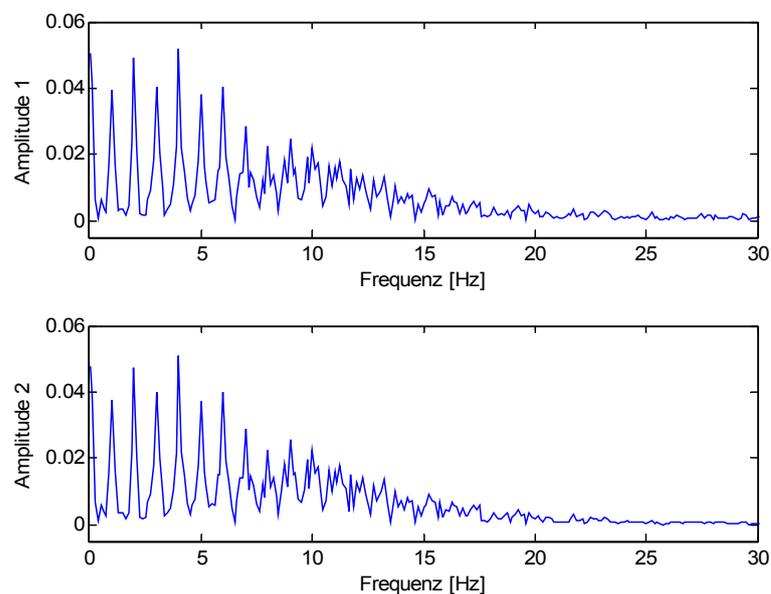


Abbildung 8.6.: Frequenzspektrum vor der Datenkompression (Amplitude 1) und nach der Signalrückgewinnung (Amplitude 2)

Die Spektren sind nahezu identisch. Man kann im oberen Frequenzbereich feststellen, dass die Signalrückgewinnung mit Upsampling und Interpolation eine geringe glättende Auswirkung auf das Signal hat. Hier sind die Amplituden der Frequenzanteile kleiner.

9. Signaldetektion

In diesem Kapitel wird die Auswertung des EKG-Signals für den Monitorkontext beschrieben. Die Auswertung hierfür beschränkt sich hauptsächlich auf die Pulsdetektion der EKG-Abschnitte. Mit der Information der Pulse über Amplitude und Zeitdauer lassen sich Signalverläufe darstellen, die dann Rückschlüsse auf krankhafte Verläufe zulassen. Dazu werden verschiedene Anwendungsbeispiele wie Herzrhythmusstörungen und Herzinfarkt generiert und die daraus resultierenden Signalverläufe beschreiben. Die Auswirkung fehlerhafter Datenübertragung auf die Signalverläufe schliessen dieses Kapitel ab.

9.1. Algorithmus

Um eine sichere Pulsdetektion auf EKG-Signale vornehmen zu können, benötigt man einen Algorithmus, der die Charakteristika des EKG-Abschnittes ausnutzt. Der signifikanteste Abschnitt ist der QRS-Komplex des Signals. Der Komplex besteht aus zwei negativen Pulsen und dem relativ steil verlaufenden R-Puls dazwischen. Ein Detektor hierfür wird in der Literatur [Krishnan 2007] vorgeschlagen. Darüber hinaus müssen zur Auswertung alle Pulse des EKG-Abschnittes ermittelt werden. Ein solch zusammengesetzter Detektor aus QRS-Komplex und P-,Q-,R-,S-,T-Pulsdetektion zeigt Abbildung 9.1.

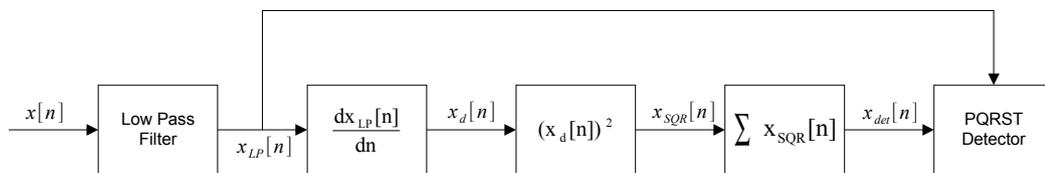


Abbildung 9.1.: Blockschaltbild zur Detektion der P-,Q-,R-,S-,T-Pulse

Das Eingangssignal $x[n]$ passiert zuerst einen Tiefpassfilter (Low Pass Filter), um das Signal zu glätten und so für die Detektion vorzubereiten. Am Ausgang des Tiefpassfilters gabelt sich das Signal $x_{LP}[n]$ auf. Zum einen führt es direkt auf die P-,Q-,R-,S-,T-Detektion und zum anderen läuft es durch den QRS-Komplex-Detektor. Dieser besteht aus einem Differentiator $\frac{dx_{LP}[n]}{dn}$, der das Signal mathematisch ableitet, einem Quadrierer $x_d^2[n]$, der kleine Signaländerungen abschwächt und grössere verstärkt. Der Integrator $\sum x_{SQR}[n]$ am

9. Signaldetektion

Ende erzeugt Mittelwerte über den Signalverlauf und unterdrückt somit höherfrequente Signalanteile. Das so entstandene QRS-Detektorsignal ist in Abbildung 9.2 unten zu betrachten.

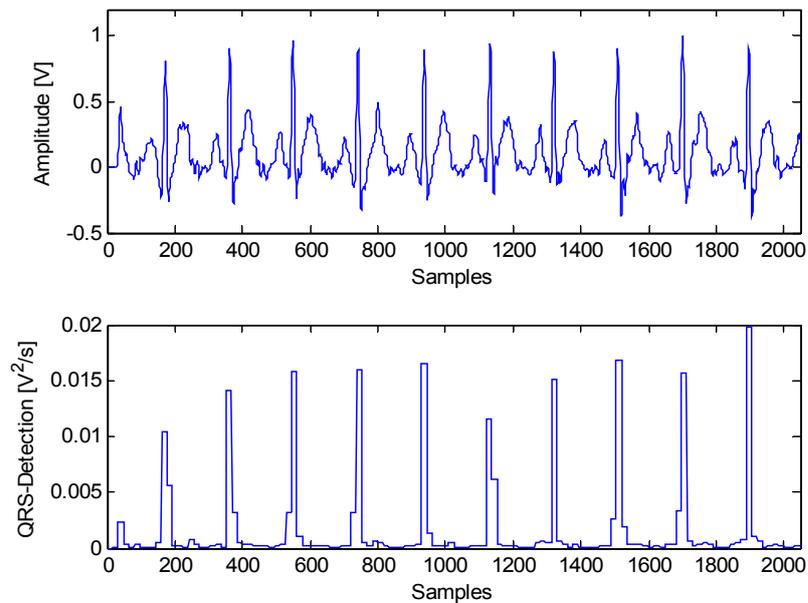


Abbildung 9.2.: EKG-Signal (oben) und das Ausgangssignal des QRS-Detektors (unten)

Das EKG-Eingangssignal $x[n]$ ist in Abbildung 9.2 oben zu sehen. Der erste Peak stellt einen unvollständig aufgezeichneten EKG-Abschnitt dar. Der QRS-Detektor „blendet“ diesen Puls aus, indem das Detektor-Signal unterhalb einer Schwelle von $0,005 \frac{V^2}{s}$ bleibt. Der P-,Q-,R-,S-,T-Detektor in Abbildung 9.1 ermittelt die Pulse eines EKG-Abschnittes, indem er den R-Puls misst und diesen mit dem QRS-Detektorsignal vergleicht. Ausgehend von einem gültigen R-Puls werden davor und danach jeweils Zeitfenster aufgespannt, bei denen die Maximals und Minimals⁵² ermittelt werden. Die Herzrate in Schlägen pro Minute kann aus den Zeitdifferenzen der gültigen R-R-Zacken bestimmt werden. Ebenso interessant sind die Zeitdifferenzen der P-R-Zacken und der R-T-Zacken. Die so ermittelten Werte zeigen die Abbildung 9.3 bei einer mittleren Herzrate von 60 Schlägen pro Minute und Abbildung 9.4 bei einer mittleren Herzrate von 120 Schlägen pro Minute.

⁵²Das entspricht den Amplitudenwerte der P-,Q-,S- und T-Zacken

9. Signaldetektion

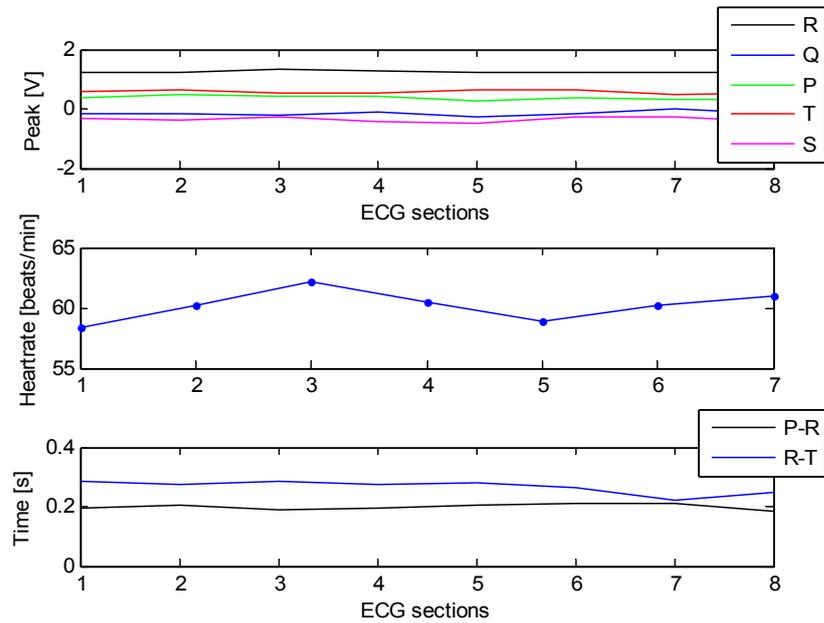


Abbildung 9.3.: Diagramm zur Detektion der P-,Q-,R-,S-,T-Pulse mit einer Herzfrequenz von 60 Schlägen pro Minute

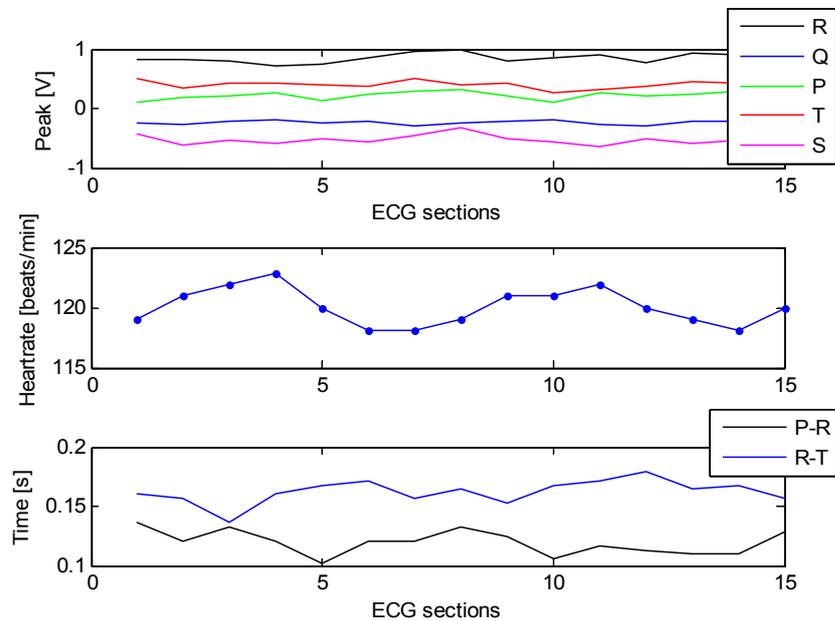


Abbildung 9.4.: Diagramm zur Detektion der P-,Q-,R-,S-,T-Pulse mit einer Herzfrequenz von 120 Schlägen pro Minute

9. Signaldetektion

Die Simulation generiert EKG-Signale mit einer einstellbaren Herzratenvariabilität von 1 Schlag pro Minute. Das oberste Diagramm von Abbildung 9.3 und 9.4 zeigt jeweils die Amplitudenverläufe der P-,Q-,R-,S-,T-Zacken eines normalen EKG-Signals. Das mittlere Diagramm zeigt die Herzratenvariabilität und das unterste die Zeitabstände der P-R- und R-T-Messung. In den folgenden Beispielen werden anhand dieser Diagramme die Auswertungen vorgenommen. Die Simulation ist darüber hinaus mit den Störanteilen (Rauschen, höherfrequente Signalanteile (50Hz) und Mittelwertschwankungen) von Kapitel 5.1 durchgeführt, um eine realistische Auswertung für die Beispiele im folgenden Kapitel vorzunehmen.

9.2. Beispiele

Die folgenden Beispiele beziehen sich auf reale Anwendungsfälle. Hierbei können die dazugehörigen EKG-Signalverläufe nur stichprobenartig besprochen werden. Die Vielfältigkeit der Herzrhythmusstörungen im EKG-Signalverlauf ist in den folgenden Beispielen beschränkt auf den AV-Block⁵³. Beim Herzinfarkt wird ebenso in diesem Rahmen nur der Vorderwandinfarkt⁵⁴ behandelt. Bei der fehlerhaften Datenübertragung werden bei den Beispielen zwischen einem einfachen Bitfehler und mehrfachen Bitfehlern innerhalb eines Datenblocks unterschieden.

9.2.1. Herzrhythmusstörungen

Die Herzrhythmusstörungen sind sehr unterschiedlich in ihrem Erscheinungsbild. Wie bereits im vorigen Abschnitt erwähnt, wird an dieser Stelle der AV-Block (atrioventrikulärer Block) behandelt. Man unterscheidet hierbei zwischen Grad 1, 2 und 3 [So 2001]. Der Grad 1 kann sowohl beim Herzgesunden als auch beim Herzkranken auftreten. Der Grad 2 ist eine schwere Rhythmusstörung und unterscheidet sich wiederum zwischen Typ 1 und Typ 2. Beim Grad 3 besteht keine Beziehung mehr zwischen Vorhof- und Kammererregung und stellt somit auch eine schwere Herzkrankheit dar. In Abbildung 9.5 oben ist das EKG-Signal eines AV-Blocks zweiten Grades vom Typ 2 zu sehen.

Im EKG-Bild sind zwar die P-Zacken sichtbar aber die zugehörigen QRS-Komplexe fehlen (Ellipsen der Pfeile 1 und 2 in der Abbildung 9.5). In der Auswertung führt es dann dazu, dass die Herzrate aufgrund des fehlenden R-Pulses plötzlich nach unten auf etwa den halben Wert springt. In Abbildung 9.5 ist das mit den Pfeilenden von 1 und 2 gekennzeichnet. Beim AV-Block Grad 3 hat die Vorhof- und die Kammererregung jeweils ihren eigenen

⁵³Der AV-Block ist eine Störung der Erregungsleitung zwischen Vorhof und Kammer.

⁵⁴Der Vorderwandinfarkt tritt durch Verschluss der linken Koronararterie und/oder ihrer Äste auf.

9. Signaldetektion

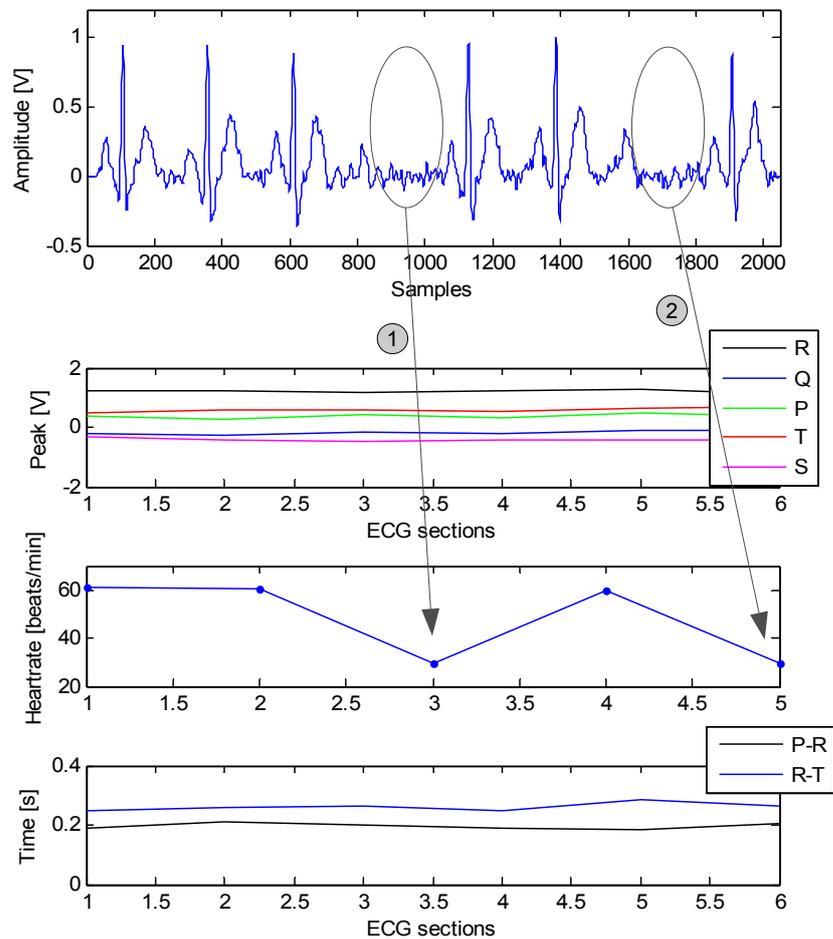


Abbildung 9.5.: EKG-Signal der Herzrhythmusstörung AV-Block 2. Grades Typ 2 (oben) und die Auswertung (unten)

Rhythmus. Das bedeutet, dass der P-Puls beliebig im EKG-Signal auftreten kann. Diesen Umstand zeigt das EKG-Signal in Abbildung 9.6 oben.

Im Kreis von Pfeil 1 ragt die P-Zacke schon recht weit in den QRS-Komplex hinein. Die Q-Zacke wird positiv und die P-R-Zeit verkürzt sich. Beim Kreis von Pfeil 2 befindet sich der P-Puls mittlerweile zwischen R- und T-Zacke. Die S-Zacke wird positiver und der Amplitudenwert vor der R-Zacke sinkt. Beim Kreis von Pfeil 3 ist der P-Puls aus diesem EKG-Abschnitt völlig verschwunden. Der Amplitudenwert vor dem R-Puls ist sehr klein (geht gegen 0). Der Kreis im EKG-Abschnitt beim Pfeil 4 ähnelt dem vom Pfeil 1, auch hier ist die P-Zacke dem QRS-Komplex nähergerückt, erkennbar durch die positive Q-Zacke und der verkürzten P-R-Zeit.

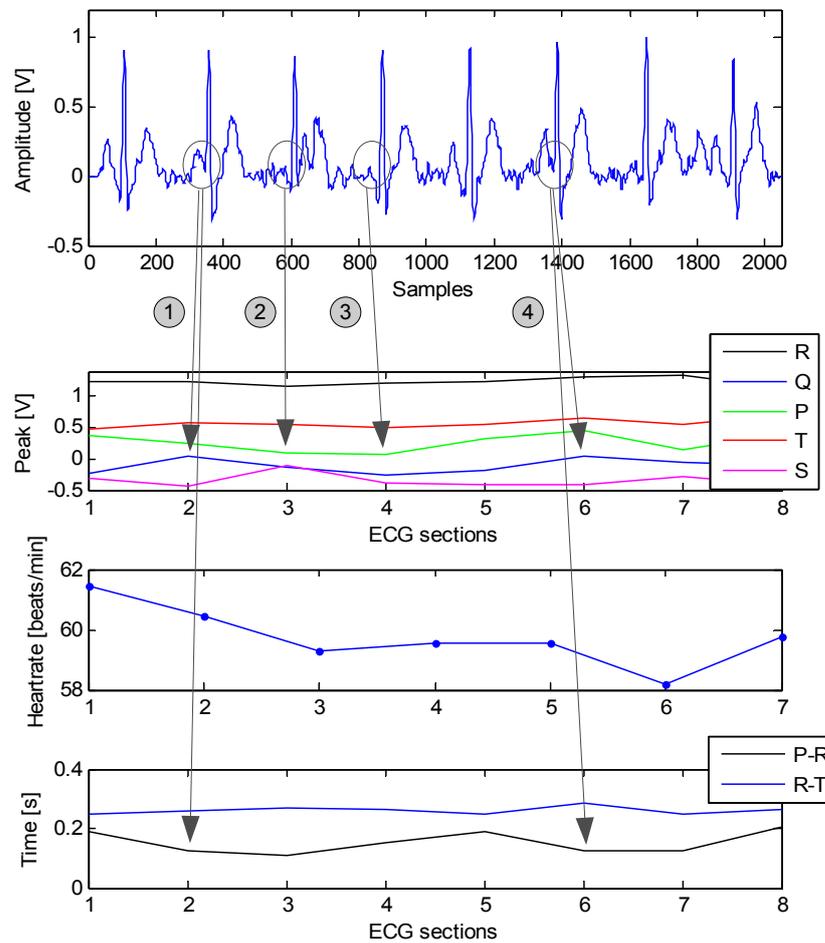


Abbildung 9.6.: EKG-Signal der Herzrhythmusstörung AV-Block 3. Grades (oben) und die Auswertung (unten)

9.2.2. Vorderwandinfarkt

Beim frischen bzw. akuten Vorderwandinfarkt entsteht der Infarkt in der vorderen Wand der linken Herzkammer. Ein Infarkt der rechten Herzkammer ist extrem selten [So 2001]. Beim Vorderwandinfarkt entsteht eine deutliche ST-Anhebung. Diese ST-Anhebung ist im EKG-Signal in Abbildung 9.7 oben am Pfeilansatz von Pfeil 1 zu sehen. Man erkennt, dass die S-Zacke positiv wird und die T-Zacke nähert sich von der Amplitude her der R-Zacke.

In der Auswertung am Pfeilende von Pfeil 1 ist die ST-Anhebung deutlich zu sehen.

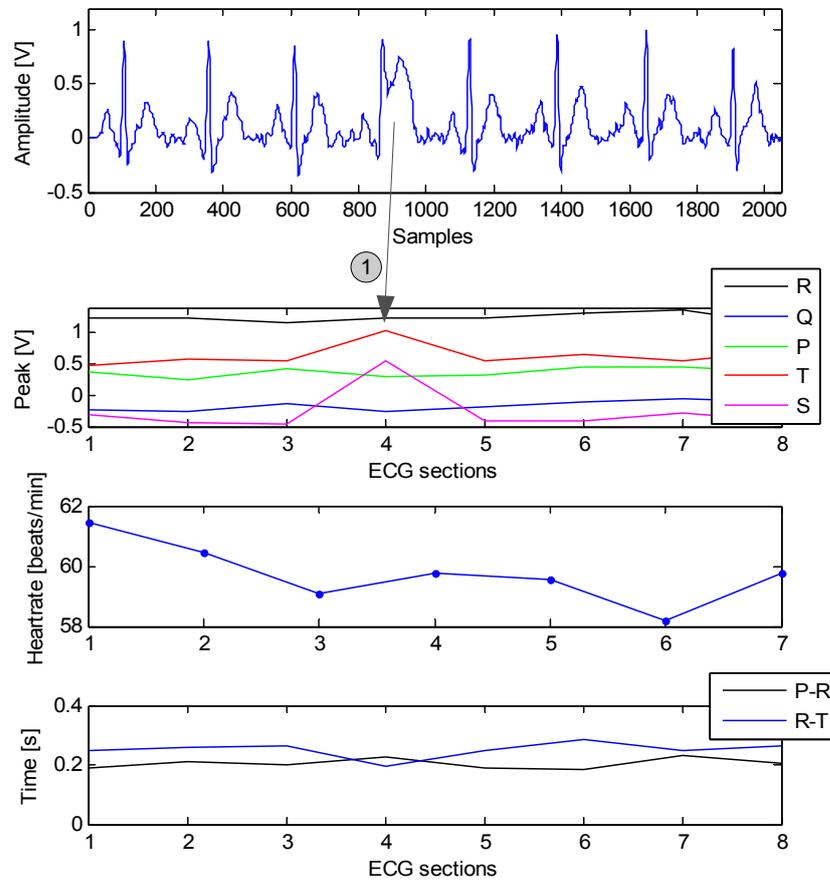


Abbildung 9.7.: EKG-Signal des Vorderwandinfarkts (oben) und die Auswertung (unten)

9.2.3. Fehlerhafte Übertragung

Die bisherigen Betrachtungen bezogen sich auf krankhafte Veränderungen im EKG-Signal. In diesem Kapitel soll die Auswirkung von Übertragungsfehlern auf die Auswertung untersucht werden. Dazu werden im Simulator Bitfehler erzeugt. Mittels der Decodierstrategie 3 in Kapitel 8.2 kann der detektierte Bitfehler mittels Korrekturwertes ersetzt werden. Abbildung 9.8 zeigt in der linken Abbildung den detektierten Bitfehler mit dem Wert 255 und auf der rechten Abbildung den Korrekturwert.

9. Signaldetektion

Decoded Table			Estimated Table		
	transmitted	received		transmitted	received
139	6	6	139	6	6
140	2	2	140	2	2
141	12	255	141	12	-2
142	2	2	142	2	2
143	0	0	143	0	0

Abbildung 9.8.: Simulatorengabe zur Erkennung von Bitfehlern (links) und zur Schätzung des Korrekturwertes (rechts)

Die farbliche Kennzeichnung hat 3 Kategorien:

- **Grün:** Der Übertragungsfehler wurde erkannt und der Korrekturwert stimmt mit dem ursprünglichen Wert überein.
- **Gelb:** Der Übertragungsfehler wurde erkannt, aber der Korrekturwert weicht vom ursprünglichen Wert ab.
- **Rot:** Der Übertragungsfehler wurde nicht erkannt.

Bei der Datenübertragung in Abbildung 9.8 wurde ein einfacher Bitfehler erzeugt. Abbildung 9.9 zeigt die Auswertung zu der fehlerhaften Datenübertragung. Es ist keine offensichtliche Auswirkung erkennbar.

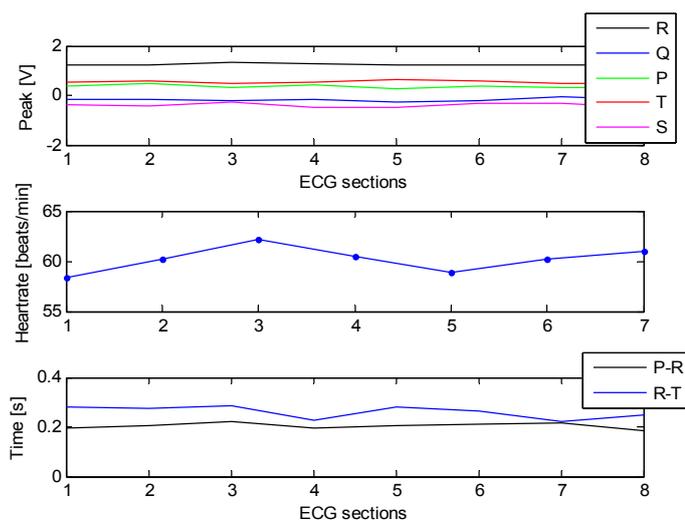


Abbildung 9.9.: Auswirkung des Bitfehlers auf die Auswertung

9. Signaldetektion

Zur weiteren Untersuchung wird eine Datenübertragung mit mehreren verschiedenen Bitfehlern erzeugt. Die folgende Abbildung 9.10 zeigt einen Ausschnitt der Simulation.

	transmitted	received
146	-4	-4
147	0	0
148	-2	-2
149	-2	-2
150	-8	-8
151	-2	-2
152	30	16
153	36	36

Abbildung 9.10.: Simulation einer fehlerhaften Datenübertragung mit mehreren Bitfehlern

Zu sehen ist am grün markierten Feld, dass ein Korrekturwert berechnet wurde, der dem ursprünglichen Wert gleicht. Die Tabelle 9.1 zeigt alle Stellen der Übertragung, an denen Bitfehler aufgetreten sind.

Tabelle 9.1.: Tabelle zur Darstellung der Übertragungsfehler und Schätzung der Korrekturwerte

Index	Anzahl Bitfehler	Referenzwert	Korrekturwert
56	2	2	2
109	5	6	2
147	1	0	0
152	4	30	16
182	3	0	0
194	4	-4	-4
240	1	-4	4
259	3	-4	-4
273	3	4	-12
284	2	2	-8
295	4	2	12
331	5	0	0
384	2	0	0
436	5	-2	-2
439	1	-2	-2

9. Signaldetektion

Die rot markierten Korrekturwerte zeigen Stellen, an denen Fehler gehäuft aufgetreten sind und deren Korrekturwerte auch stärker von ursprünglichen Werten abweichen. Die Auswirkungen auf die Auswertung zeigt Abbildung 9.11. Bereiche, in denen Fehler vereinzelt auftreten scheinen der Auswertung wenig bzw. gar nichts auszumachen. Treten jedoch Fehler gehäuft auf, wie es im Bereich der Indizes 273, 284 und 295 der Fall ist, so stellt man Auswirkungen in der Auswertung fest. Die Pfeile in der Abbildung 9.11 mit 1 gekennzeichnet zeigen auf einen erhöhten T-Puls und einen kleineren Q-Puls im EKG-Abschnitt mit der Nummer 5.

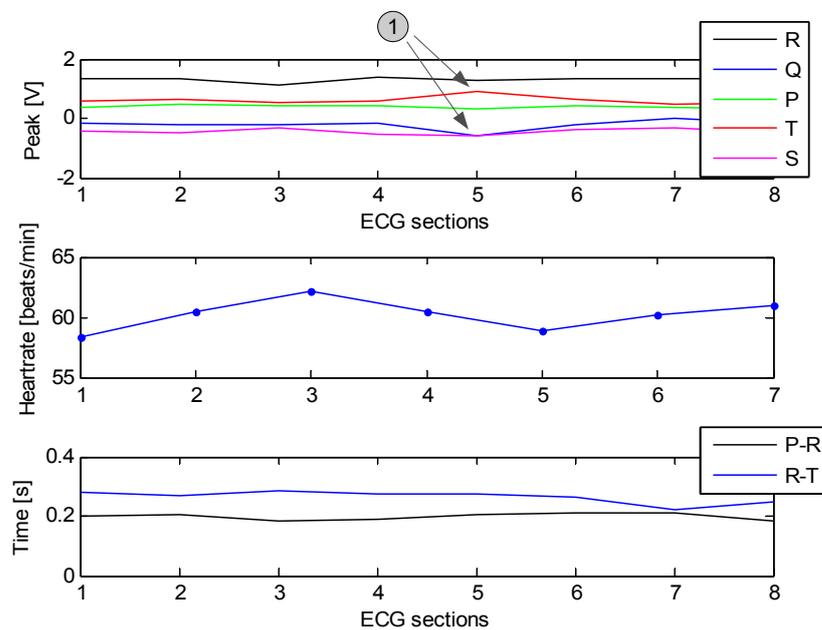


Abbildung 9.11.: Auswirkung der Bitfehler auf die Auswertung

9.3. Resümee

Die in diesem Kapitel beschriebenen Anwendungsfälle von krankhaften EKG-Signalen stellen nur einen Auszug aller bekannten Krankheitsbilder dar. Es hat sich jedoch gezeigt, dass der für die Beispiele angewendete Algorithmus sich gut eignet, um krankhafte Verläufe im EKG-Signal hervorzuheben. Zur Detektion von Krankheitsbildern müssen jedoch genaue Kriterien festgelegt werden, dies würde den Rahmen dieser Arbeit sprengen. Bei Übertragungsfehlern mit wenigen vereinzelt Bitfehlern sind Auswirkungen auf die Auswertung kaum feststellbar. Dies liegt auch darin begründet, dass die Korrekturwerte nach der Decodierstrategie 3 teilweise die Fehler komplett ausgleichen. Treten die Fehler aber gehäuft

9. Signaldetektion

auf, so sind schon Auswirkungen auf die Auswertung feststellbar. Der Benutzer sollte auf jeden Fall informiert werden, wenn Fehler bei der Übertragung aufgetreten sind. Er kann dann gegebenenfalls die Übertragung nochmals ausführen und sicherstellen, dass eventuell etwaige Störquellen abgeschaltet sind, oder aber einfach die Distanz zum Empfänger verringern, um ein größeres Signal-Rausch-Verhältnis zu erreichen.

10. Zusammenfassung und Ausblick

Im Rahmen dieser Masterarbeit wurde eine Konzeption für einen drahtlosen EKG-Monitor erarbeitet. Der drahtlose EKG-Monitor hat die Aufgabe, das Elektrokardiogramm aufzunehmen, zur weiteren Verarbeitung aufzubereiten und das Signal komprimiert zu einer Gegenstelle zu übertragen. Die Gegenstelle hierbei dient als Gateway, um das Elektrokardiogramm über das Internet zu verschicken. Die mobile Einheit befindet sich am Patienten, der sich dann dadurch frei bewegen kann.

Zur Überprüfung auf Realisierbarkeit der verwendeten Methoden und Techniken ist eine Matlab-Simulation für die Signalverarbeitung und ein Qt-Simulator für die Entropie- und Kanalcodierung entstanden.

Zunächst wurde der Systemkontext der Anwendung betrachtet. Die für die Anwendung benötigte Funkverbindung zur Übertragung des Elektrokardiogramms wurde zwischen Bluetooth und einer einfachen unidirektionalen Funkverbindung abgewogen. Durch eine Nutzen-Aufwand Betrachtung fiel die Entscheidung auf Letztere. Für die Funkverbindung zur Notrufabsetzung wurde ein Konzept mit GSM-Modul und SIM-Karte diskutiert.

Für die Signalaufbereitung kommen Filter der Signalverarbeitung zum Einsatz, die Störungen des EKG-Signals beseitigen bzw. vermindern sollen. Damit das Rauschen des Signals vermindert wird, kommt ein nichtlineares Filter mit morphologischen Operatoren zur Anwendung. Um Mittelwertschwankungen auszugleichen hilft ein Medianfilter diese auszugleichen und sorgt somit für ein schwankungsfreies Signal. Gegen höherfrequente Störsignale wird ein Tiefpassfilter so dimensioniert, dass es oberhalb der EKG-Grenzfrequenz alle Frequenzen dämpft.

Zu einer effektiven Datenübertragung gehört es sich, dass die Daten komprimiert übertragen werden. Um dies bewerkstelligen zu können, kommen auch hier verschiedene Techniken zur Anwendung. Zuerst hilft das Heruntertakten (Downsampling), die Datenrate zu verkleinern. Mit dem nachfolgenden Prädiktor erhält man ein dekorreliertes Signal das eine Reduktion von 2Bit pro Zeichen ohne große Verluste zulässt. Um die Datenrate noch weiter zu reduzieren kommt eine Huffman-Codierung zum Einsatz. Mit dieser Codierung ist es möglich, häufig auftretende Zeichen mit einer kürzeren Bitlänge zu codieren. Man kommt

10. Zusammenfassung und Ausblick

somit schliesslich auf eine Kompression von 85% der ursprünglichen Datenmenge.

Bei der Übertragung kann es jedoch auch zu Störungen kommen, dies kann zu Bitfehlern der übertragenen Daten führen. Damit die Fehler erkannt werden können, wurde ein verkürzter zyklischer Blockcode mit der Kurzbezeichnung (8,3,4)-Blockcode entwickelt. Dieser Code hat die Fähigkeit, Bündelfehler zu detektieren.

Nach dem Empfang des Signals auf der Gegenstelle und der notwendigen Decodierungen kann das Signal wieder zurückgewonnen werden. Zu Zwecken weiterer Auswertungen und zum Erstellen von Signalverläufen der EKG-Abschnitte ist es hilfreich eine Detektion der EKG-Pulse vorzunehmen. Dazu wurden Detektoren für den QRS-Komplex und für die EKG-Pulse implementiert. Abschließend erfolgt eine Auswertung des Elektrokardiogramms mit Hilfe der implementierten Detektoren anhand von Elektrokardiogramm-Beispielen. Die Beispiele setzen sich hauptsächlich aus Herzkrankheiten wie Herzrhythmusstörungen und Vorderwandinfarkt zusammen.

Weiterführend kann aus der in dieser Masterarbeit entwickelten Konzeption ein reales System entwickelt werden. Als Grundlage dafür dienen zur C bzw. C++ Implementierung die Komponentendiagramme A.3 und A.4, im Anhang A.8. Diese Diagramme zeigen die zu implementierenden Module bzw. Klassen in der mobilen Einheit des Monitors in Abbildung A.3 und der Gegenstelle in Abbildung A.4.

Die Übertragung der EKG-Informationen über das Internet vom Gateway aus konnte nur sehr oberflächlich behandelt werden. Für die Übertragung existieren E-mail Protokolle wie SMTP, die sich hierfür einsetzen lassen. Dazu kann auch auf verfügbare embedded Webserver zurückgegriffen werden, die frei im Handel verfügbar sind.

Für das Absetzen eines Notrufes über das GSM-Netz existieren Telefonmodule u.a. von Siemens, die hierfür eingesetzt werden können. Diese Module lassen sich steckbar in die Anwendung integrieren. Angesprochen werden die Module über AT-Befehle.

Literaturverzeichnis

BiPhaseMark 2009

BiPhase Mark Code (Primärquelle: Universität Oldenburg). Abrufdatum: Juni 2009.
<http://einstein.informatik.uni-oldenburg.de/rechnernetze/bi-phase1.htm>
7

Bossert 1998

BOSSERT, Martin: *Kanalcodierung*. B. G. Teubner Stuttgart, 1998. – ISBN 3-519-16143-5 20

Bundesamt 2009

Herzerkrankungen in Deutschland (Primärquelle: Statistisches Bundesamt). Abrufdatum: Juni 2009. <http://www.gbe-bund.de> 3

Carl 2009

CARL, Prof. Dr. H.: *Informationstheorie und Codierung*. Georg-Simon-Ohm Hochschule Nürnberg, 2009 20, 60, 61, 62, 65

EKGEntstehung 2009

Entstehung des EKG Signals (Primärquelle: Wikipedia). Abrufdatum: Juli 2009. <http://de.wikipedia.org/wiki/Elektrokardiogramm> VIII, 24

EKGGenerator 2009

A realistic ECG Waveform Generator (Primärquelle: Physionet). Abrufdatum: Mai 2009. <http://www.physionet.org/physiotools/ecgsyn/> 28

EKGOnline 2009

Erweiterte Ableitungen (Primärquelle: EKG-Online). Abrufdatum: Mai 2009. <http://www.grundkurs-ekg.de> 4

FIPM 2009

EKG Folie (Primärquelle: Fraunhofer-Institut Photonische Mikrosysteme). Abrufdatum: Juni 2009. <http://www.ipms.fraunhofer.de/> VIII, 15, 16

Grueningen 2002

GRUENINGEN, Daniel C.: *Digitale Signalverarbeitung*. Carl Hanser Verlag, 2002. – ISBN 3-446-21976-5 IX, 16, 39, 43

Herter 2004

HERTER, Eberhard: *Nachrichtentechnik*. Hanser Verlag, 2004. – ISBN 3-4462-2684-2
27

IfBTUK 2009

Eindhoven Ableitungen (Primärquelle: Institut für Biomedizinische Technik Universität Karlsruhe). Abrufdatum: Juli 2009. www.cvrtil.utah.edu/~fs/lessons/messtechnik/Vorlesung8.pdf VIII, 14, 15

Krishnan 2007

KRISHNAN, S.M.: *Advances in Cardiac Signal Processing*. Springer-Verlag, 2007. – ISBN 3-540-36674-1 VIII, IX, 15, 32, 33, 34, 38, 39, 40, 41, 77

Merkle 2002

MERKLE, Andreas: *Digitale Funkkommunikation mit Bluetooth*. Franzis Verlag, 2002. – ISBN 3-7723-4654-5 VIII, XI, 5, 6, 7

Roppel 2006

ROPPEL, Carsten: *Grundlagen der digitalen Kommunikationstechnik*. Hanser Verlag, 2006. – ISBN 3-446-22857-8 59

So 2001

So, Cook-Sup: *Praktische EKG-Deutung*. Georg Thieme Verlag, 2001. – ISBN 3-13-112683-3 2, 14, 21, 23, 25, 80, 82

Steiger 2009

Notfallhilfe (Primärquelle: Björn-Steiger-Stiftung). Abrufdatum: Juni 2009. <http://www.steiger-stiftung.de/> 9

Strutz 2005

STRUTZ, Tilo: *Bilddatenkompression*. Friedr. Vieweg & Sohn Verlag, 2005. – ISBN 3-528-23922-0 VIII, 12, 17, 18

Werner 2006

WERNER, Martin: *Nachrichten-Übertragungstechnik*. Vieweg Verlag, 2006. – ISBN 3-528-04126-9 58

Werner 2008

WERNER, Martin: *Digitale Signalverarbeitung mit MATLAB-Praktikum*. Friedr. Vieweg & Sohn Verlag, 2008. – ISBN 978-3-8348-0393-1 IX, 34, 36, 37, 48, 51, 74

A. Anhang

A.1. Listing zum EKG-Signalspektrum

Listing A.1: Berechnung EKG-Signalspektrum in Matlab

```

% values used from ECG Signal specification range
UR          = 0.0014;      %Amplitude R-Pulse
UQS         = 0.00028;    %Amplitude Q- and S-Pulse
UPT         = 0.0002;    %Amplitude P- and T-Pulse
n           = 0.0:0.01:100;

Tptl        = 0.085;      %Time (P or T)/2
Tsl         = 0.21;      %Time between P or T and R
Tkl         = 0.005;      %Time (Q or S)/2
Til         = 0.025;      %Time R/2
Tjl         = 0.03;      %Time between Q or S and R
Tl          = 1.0;       %Pulse rate 60

Tpth        = 0.05;      %Time (P or T)/2
Tsh         = 0.1;       %Time between P or T and R
Tkh         = 0.003;      %Time (Q or S)/2
Tih         = 0.015;      %Time R/2
Tjh         = 0.018;      %Time between Q or S and R
Th          = 0.5;       %Pulse rate 120

spektrum_Rposl = UR*2*(Til/Tl)*((sin(pi*n*(Til/Tl))./(pi*n*(Til/Tl))).^2);
spektrum_Rnegl = UQS*2*(Tkl/Tl)*2*cos(2*pi*n*(Tjl/Tl)).*((sin(pi*n*(Tkl/Tl))./(pi*n*(Tkl/Tl)))
    .^2);
spektrum_PTI   = UPT*2*(Tptl/Tl)*2*cos(2*pi*n*(Tsl/Tl)).*((1-((2*n*(Tptl/Tl)).^2)).^(-1)).*(sin
    (2*pi*n*(Tptl/Tl))./(2*pi*n*(Tptl/Tl)));
spektrum_Gesl  = spektrum_PTI+spektrum_Rposl-spektrum_Rnegl;

spektrum_Rposh = UR*2*(Tih/Th)*((sin(pi*n*(Tih/Th))./(pi*n*(Tih/Th))).^2);
spektrum_Rnegh = UQS*2*(Tkh/Th)*2*cos(2*pi*n*(Tjh/Th)).*((sin(pi*n*(Tkh/Th))./(pi*n*(Tkh/Th)
    ))).^2);
spektrum_PTh   = UPT*2*(Tpth/Th)*2*cos(2*pi*n*(Tsh/Th)).*((1-((2*n*(Tpth/Th)).^2)).^(-1)).*(
    sin(2*pi*n*(Tpth/Th))./(2*pi*n*(Tpth/Th)));
spektrum_Gesh  = spektrum_PTh+spektrum_Rposh-spektrum_Rnegh;

figure (1)
clf

subplot (211)
plot (n,spektrum_Gesl,'-')
ylim([-0.00005 0.0002])
title ('Model Spektrum - 60 Beats per minute','FontName','Arial')

```

A. Anhang

```
xlabel ('Frequenz [Hz]', 'FontName', 'Arial')
ylabel ('Amplitude [V]', 'FontName', 'Arial')
%grid on
set(gca, 'FontName', 'Arial')

subplot (212)
plot (n, spektrum_Gesh, '-')
ylim([-0.00005 0.0002])
title ('Model Spektrum - 120 Beats per minute', 'FontName', 'Arial')
xlabel ('Frequenz [Hz]', 'FontName', 'Arial')
ylabel ('Amplitude [V]', 'FontName', 'Arial')
%grid on
set(gca, 'FontName', 'Arial')
%set(gca, 'XGrid', 'off');
```

A.2. Listing zum Mittelwertfilter

Listing A.2: Mittelwertfilter in Matlab

```
function [y_median_out] = medianfilter( x_median_in )

% get data length (2048) -> 8sec
x_len = length(x_median_in);

% 32 bytes -> 125ms
% 4 * 32 bytes = 128 bytes -> 500ms

% initialisation of used buffers
mean_vector = zeros(4,1);
y_median_out = x_median_in;

% setup loop variables
loopi = x_len/128;
x_index = 1;

% remove mean value variations
for i = 1:loopi
    x_index_old = x_index;
    for j = 1:4
        mean_vector(j) = sum(x_median_in(x_index:x_index+31))/32;
        x_index = x_index + 32;
    end
    min_offset = min(mean_vector);
    y_median_out(x_index_old:x_index-1) = y_median_out(x_index_old:x_index-1)-min_offset;
end

figure (22)
clf

subplot(211)
plot(x_median_in,'-')

subplot(212)
plot(y_median_out,'-k')

end
```

A.3. Listing zum Levinson-Durbin Algorithmus

Listing A.3: Levinson-Durbin Algorithmus zur Berechnung der Reflexionskoeffizienten in C

```

#include <math.h>
#include "levinsondurbin.h"

/*****
/*                                     */
/*           Levinson-Durbin-Algorithmus           */
/*                                     */
/* *k:   OUT Pointer to reflection coefficients   */
/* *r:   IN  Pointer to Autocorrelation function   */
/* p:   IN  Prediction order (max 50)           */
/*      RET true if successfull otherwise false   */
*****/

int levinsondurbin( double *k, double *r, int p )
{
    double alpha [50][50];
    double E[50];
    double q [50];
    int j, m;

    /* Parameter check */
    if ( (p < 2) || (p > 50) )
    {
        return 0;
    }

    /* initialisation with zeros */
    for ( j = 0; j < 50; j++ )
    {
        memset(&alpha[j][0], 0.0, 50);
    }
    memset(E, 0.0, 50);
    memset(q, 0.0, 50);
    memset(k, 0.0, 50);

    /* initialisation with start values */
    alpha [0][0] = -((*r+1)/(*r+0));
    E[0] = (*r+0) + (alpha[0][0]*(*r+1));
    q[0] = (*r+2) + (alpha[0][0]*(*r+1));

```

A. Anhang

```
*(k+0) = alpha[0][0];

for( m = 1; m < p; m++ )
{
    int i, n;
    *(k+m) = -(q[m-1]/E[m-1]);
    E[m] = E[m-1]*(1-pow(*(k+m),2.0));
    for( i = 0; i <= (m-1); i++ )
    {
        alpha[m][i] = alpha[m-1][i] + (*(k+m))*alpha[m-1][m-i];
    }
    alpha[m][m] = *(k+m);
    q[m] = *(r+m+2);
    for( n = 0; n <= m; n++ )
    {
        q[m] = q[m] + (alpha[m][n]*(*(r+m-n+2)));
    }
}

return 1;
}
```

A.4. Listing zur Rauschunterdrückung

Listing A.4: Rauschunterdrückung mittels morphologischen Operatoren in Matlab

```

function [mean_value] = noisefilter( x_in )

% get data length
x_in_len = length(x_in);

% initialize all buffer with zeros
FilterMax = zeros(x_in_len,1);
FilterMaxMin = zeros(x_in_len,1);
FilterMin = zeros(x_in_len,1);
FilterMinMax = zeros(x_in_len,1);
mean_value = zeros(x_in_len,1);

% initialize Mask (structure components)
Mask1 = [0 0.004 0.02 0.004 0];
Mask2 = [0 0 0 0 0];

% startup Max, MaxMin, Min and MinMax buffer
FilterMax(1) = max(x_in(1:3));
FilterMax(2) = max(x_in(1:4));
FilterMax(3) = max(x_in(1:5));
FilterMax(4) = max(x_in(2:6));
FilterMaxMin(1) = min(FilterMax(1:3));
FilterMaxMin(2) = min(FilterMax(1:4));

FilterMin(1) = min(x_in(1:3));
FilterMin(2) = min(x_in(1:4));
FilterMin(3) = min(x_in(1:5));
FilterMin(4) = min(x_in(2:6));
FilterMinMax(1) = max(FilterMin(1:3));
FilterMinMax(2) = max(FilterMin(1:4));

for i = 3:x_in_len-4
    % calculate max peaks
    masked = x_in(i:i+4)-Mask1(1:5);
    FilterMax(i+2) = max(masked);
    % calculate min of max peaks
    masked = FilterMax(i-2:i+2)-Mask2(1:5)';
    FilterMaxMin(i) = min(masked);
    % calculate min peaks
    masked = x_in(i:i+4)-Mask1(1:5);
    FilterMin(i+2) = min(masked);
    % calculate max of min peaks
    masked = FilterMin(i-2:i+2)-Mask2(1:5)';

```

A. Anhang

```

    FilterMinMax(i) = max(masked);
end

FilterMax(x_in_len-1) = max(x_in(x_in_len-3:x_in_len));
FilterMax(x_in_len) = max(x_in(x_in_len-2:x_in_len));
FilterMin(x_in_len-1) = min(x_in(x_in_len-3:x_in_len));
FilterMin(x_in_len) = min(x_in(x_in_len-2:x_in_len));

FilterMaxMin(x_in_len-1) = min(FilterMax(x_in_len-3:x_in_len));
FilterMaxMin(x_in_len) = min(FilterMax(x_in_len-2:x_in_len));
FilterMinMax(x_in_len-1) = max(FilterMin(x_in_len-3:x_in_len));
FilterMinMax(x_in_len) = max(FilterMin(x_in_len-2:x_in_len));

% calculate mean value
mean_value = FilterMaxMin+FilterMinMax;
mean_value = mean_value.*0.5;

figure (20)
clf

subplot(211)
plot(x_in,'-')
xlim([0 750]);
title('ECG Signal with noise','FontName','Arial')
xlabel('Samples','FontName','Arial')
set(gca,'FontName','Arial')

subplot(212)
plot(mean_value,'-')
xlim([0 750]);
title('ECG Signal noise reduced','FontName','Arial')
xlabel('Samples','FontName','Arial')
set(gca,'FontName','Arial')

end

```

A.5. Listing zur Skalierung und Koeffizientenquantisierung des Tiefpassfilters

Listing A.5: Skalierung und Koeffizientenquantifizierung des Tiefpassfilters in Matlab

```

%% Quantizing and scaling Prefilter coefficients

% SOS from Filter design
SOSs = SOS;
SOSq = SOS;
disp('2nd order blocks – MATLAB design')
disp(SOS)

% scaling with "Betragbegrenzung"
N = 1024;
[H1,w] = freqz(SOS(1,1:3),SOS(1,4:6),N); % frequency responses
[H2,w] = freqz(SOS(2,1:3),SOS(2,4:6),N);
[H3,w] = freqz(SOS(3,1:3),SOS(3,4:6),N);
[H4,w] = freqz(SOS(4,1:3),SOS(4,4:6),N);
MAX1 = max(abs(H1)); % scaling factors
MAX2 = max(abs(H2.*H1/MAX1));
MAX3 = max(abs(H3.*(H2/MAX2).*(H1/MAX1)));
MAX4 = max(abs(H4.*(H3/MAX3).*(H2/MAX2).*(H1/MAX1)));
SOSs(1,1:3) = SOSs(1,1:3)/MAX1;
SOSs(2,1:3) = SOSs(2,1:3)/MAX2;
SOSs(3,1:3) = SOSs(3,1:3)/MAX3;
SOSs(4,1:3) = SOSs(4,1:3)/MAX4;
fprintf('scaling factors for section 1, 2, 3 and 4: %g , %g , %g , %g\n\n',1/MAX1,1/MAX2,1/
MAX3,1/MAX4)

disp('2nd order blocks – scaling')
disp(SOSs)

% Quantizing with 8 Bit
w = 8;
% least significant bit
LSB = 2-(w-1);

% transpose original coefficients from second section order into transfer
% function
[b,a] = sos2tf(SOSs);

% Quantize coefficients
SOSq(:,1:3) = LSB*round(SOSs(:,1:3)/LSB);
SOSq(:,5:6) = LSB*round(SOSs(:,5:6)/LSB);

% transpose original coefficients from second section order into transfer

```

KONZEPTIONIERUNG EINES DRAHTLOSEN EKG-MONITORS

A. Anhang

```
% function
[bq,aq] = sos2tf(SOSq);
disp('2nd order blocks – quantized')
disp(SOSq)

% for saving setting remove "%" in line below
% save Prefilter ;

% call view tool
fvtool(bq, aq, b, a);
```

A.6. Listing zur QRS-Komplex Detektion

Listing A.6: QRS-Komplex Detektion in Matlab

```
function [qrs_detection] = qrsdetector( x_qrs_in )

x_qrs_len = length(x_qrs_in);

% initialization of used buffers
qrs_detection = zeros(x_qrs_len,1);
qrs_detection(1) = x_qrs_in(1);

for i=2:x_qrs_len
    % differentiator
    qrs_detection(i) = x_qrs_in(i)-x_qrs_in(i-1);
    % suppress low level values
    qrs_detection(i) = qrs_detection(i)^2;
end

% mean value filter to suppress short peaks
j = 1;
for i=1:x_qrs_len/16
    meanqrs = sum(qrs_detection(j:j+15))/16;
    qrs_detection(j:j+15) = meanqrs;
    j = j + 16;
end

end
```

A.7. Simulator für Code Generierung und Übertragung

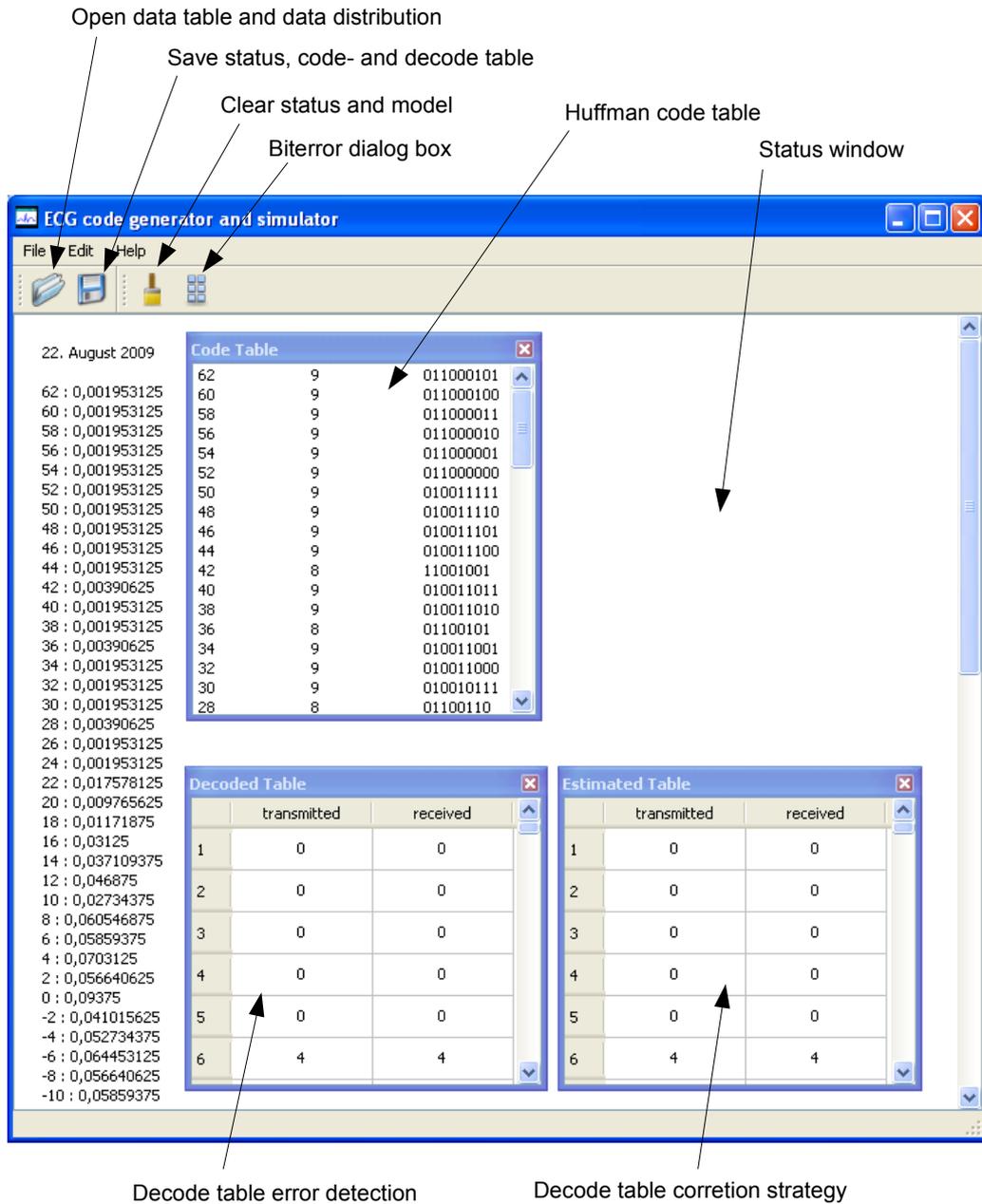


Abbildung A.1.: Haupt- und Ausgabefenster des Simulators

A.7.1. Anleitung zur Bedienung des Simulators

1. Nach dem Starten der `ecgcoder.exe` erscheint das Haupt-Fenster mit leerem Status-Fenster (Abbildung A.1).
2. Zuerst muss die csv-Datei geladen werden, die die Daten mit ihren jeweiligen Auftretenswahrscheinlichkeiten enthalten. Diese befindet sich im gleichen Ordner wie die exe-Datei und lautet *Source encoding.csv*. Nach erfolgreichem Laden erscheint die Huffman-Code-Tabelle (Huffman code table).
3. Zum Simulieren von Übertragungsfehlern steht das Biterror-Dialogfenster (Biterror dialogbox) in Abbildung A.2 zur Verfügung.

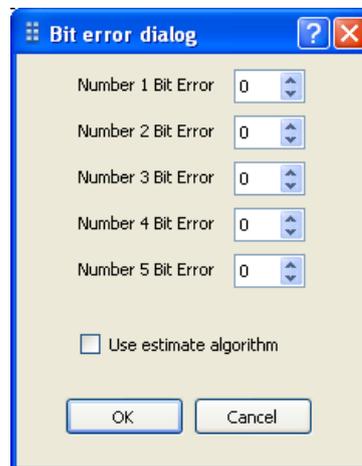


Abbildung A.2.: Biterror-Dialogfenster

In diesem Fenster können Bitfehler bis zu einer Länge von 5 Bit angegeben werden, die dann im Datenvektor an zufällig ausgewählten Stellen Fehler verursachen. Die Auswahl der Checkbox *Use estimate algorithm* darunter führt dazu, dass an den fehlerhaft erkannten Stellen der Korrekturalgorithmus nach der Decodierstrategie 3 in Kapitel 8.2 angewendet wird.

4. Zum Starten der Simulation muss eine Datendatei geöffnet werden. Dazu wird über *Open* die Datei *predictionerror.txt* geöffnet. Es öffnen sich die Fenster *Decoded Table* und *Estimated Table*. Die Fenster heben die fehlerhaft erkannten und korrigierten Stellen farblich hervor. Das Status-Fenster gibt Auskunft, an welchen Stellen die Fehler aufgetreten sind.
5. Zum Abspeichern der Simulationsdaten steht der *Save* Befehl zur Verfügung. Abgespeichert werden die Huffman-Code-Tabelle, die decodierten Daten und die Statusinformationen.

A.8. Komponentendiagramme zur Software Umsetzung

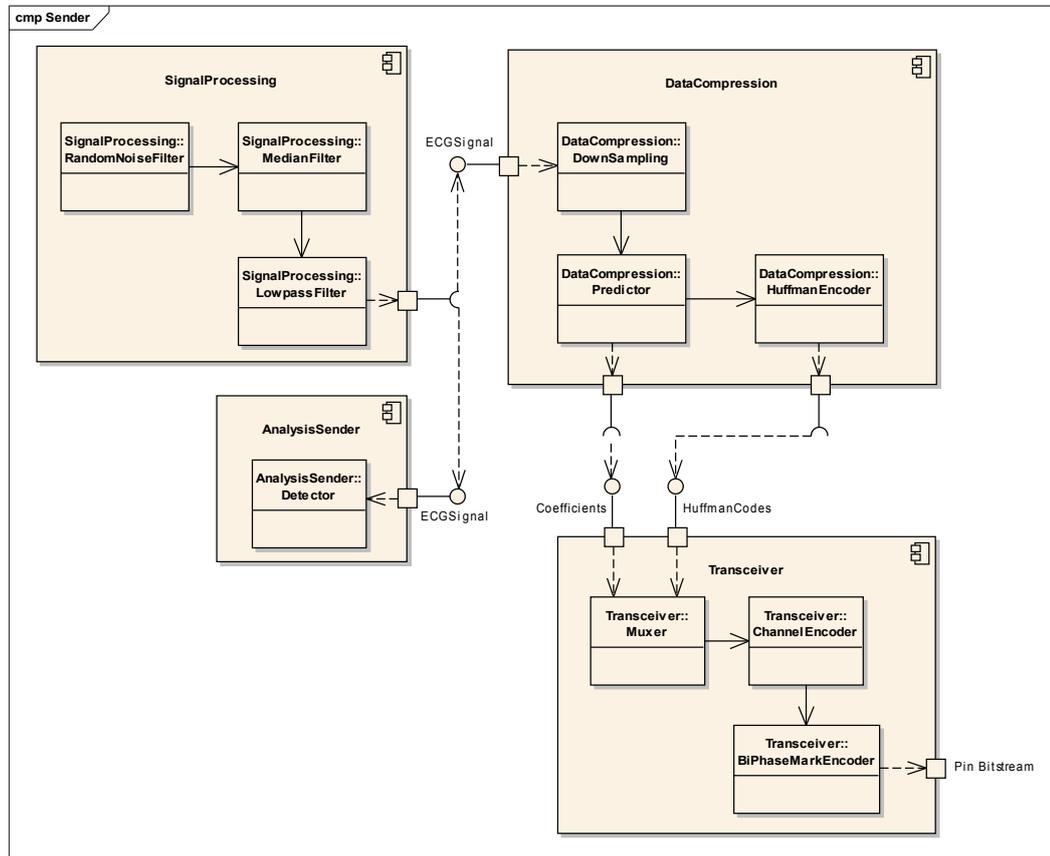


Abbildung A.3.: Komponentendiagramm für die Software Umsetzung des EKG-Monitor Sender

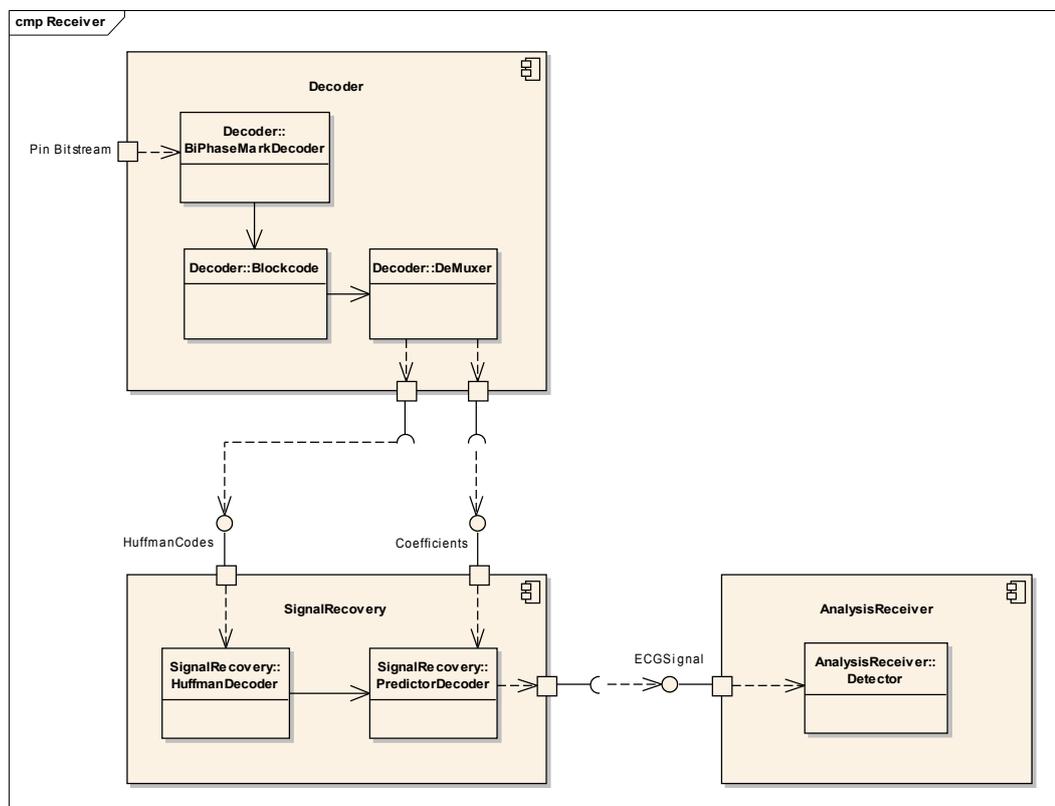


Abbildung A.4.: Komponentendiagramm für die Software Umsetzung des EKG-Monitor Empfangsteil